NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

DESIGN AND IMPLEMENTATION OF
A PERSONNEL DATABASE SYSTEM
FOR INDONESIAN NAVAL OFFICERS

by

Djoko M. Ariyadi

June 1986

Thesis Advisor: Daniel R. Dolk

Approved for public release; distribution is unlimited.

:CURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

a REPORT SECURITY CLASSIFICATION
unclassified

1b. RESTRICTIVE MARKINGS

a SECURITY CLASSIFICATION AUTHORITY

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited.

b DECLASSIFICATION / DOWNGRADING SCHEDULE

PERFORMING ORGANIZATION REPORT NUMBER(S)

5 MONITORING ORGANIZATION REPORT NUMBER(S)

a. NAME OF PERFORMING ORGANIZATION
aval Postgraduate School

6b OFFICE SYMBOL *(If applicable)*
54

7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

:. ADDRESS *(City, State, and ZIP Code)*
onterey, CA 93943-5000

7b. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943-5000

a NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

:. ADDRESS *(City, State, and ZIP Code)*

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

1 TITLE *(Include Security Classification)* UNCLASSIFIED
esign and Implementation of a Personnel Database System for Indonesian
Naval Officers

2 PERSONAL AUTHOR(S)
Djoko M. Ariyadi

3a TYPE OF REPORT
asters Thesis

13b TIME COVERED
FROM ________ TO ______

14 DATE OF REPORT *(Year, Month, Day)*
1986 June 20

15 PAGE COUNT
122

5 SUPPLEMENTARY NOTATION

7 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
database, prototype, system, normal forms

9 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The objective of this thesis is to provide a software tool to support the Deputy Chief of Staff for Personnel and Staff Function of the Indonesian Navy in making decisions with fast, timely, relevant, up-to-date and accurate information regarding Personnel Management activities. A database design is proposed including the logical and physical phases and an implementation of a personnel database prototype is undertaken on a microcomputer using dBASE II.

0 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

2a NAME OF RESPONSIBLE INDIVIDUAL
Prof. Daniel R. Dolk

22b TELEPHONE *(Include Area Code)*
408 646-2666

22c OFFICE SYMBOL
54Dk

D FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Design and Implementation of a Personnel Database System for Indonesian Naval Officers

by

Djoko M. Ariyadi
Major, Indonesian Marines
B.S., Indonesian Naval Academy, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
June 1986

ABSTRACT

The objective of this thesis is to provide a software tool to support the Deputy Chief of Staff for Personnel and Staff Function of the Indonesian Navy in making decisions with fast, timely, relevant, up-to-date and accurate information regarding Personnel Management activities.

A database design is proposed including the logical and physical phases and an implementation of a personnel database prototype is undertaken on a microcomputer using dBase II.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to acknowledge the help of those who made this thesis possible. First, of course there are Professor Daniel Roy Dolk and Professor Richard Allin McGonigal, both of whom showed great understanding during the time that it took to finish the thesis. I am also very grateful to Captain Frederick W. Weber,Jr., US Army and Lieutenant Mark L. Scire, US Coast Guard assisting in the editing of this thesis.

I. INTRODUCTION

The most important resource in any organization is its personnel. All aspects of an organization are dependent on personnel, so a large measure of the success of an organization depends on properly managing it's human resources. This makes personnel management a critical task.

Decisions which affect personnel must be carefully planned. Personnel are human beings, unlike other resources such as materiel or money. Materiel may be moved about without the same considerations involved with moving people. Materiel does not need a life-cycle management system as complex as personnel management.

The personnel administration cycle consists of procurement, classification, education, utilization, treatment, and separation [Ref. 1]. Each phase must be carefully planned and followed by a decision that requires information suppot. The information support must be accurate, timely and relevant.

Human memory is limited and does not always process information fast enough. A computer can aid the human mind because it has the capability to process large amounts of data with high accuracy and little error.

In light of the needs and capabilities described above, this thesis will discuss personnel management base on database management using microcomputers as decentraliced activities. This thesis will investigate the use of a microcomputer for processing personnel management data. It will show how a decentralized database system can provide information to assist a Deputy of Personnel and Staff Function in decision making regarding personnel management activities.

This system, which keeps track of an officer's career (job placement, promotion, education, training, etc), provides current information regarding an officer's career and shows what kind of jobs or education/training should be scheduled for a certain officer to gain required knowledge and experience before being assigned a specific job.

A reorganization of the Indonesian Navy was applied at the end of 1985. Because of the restructuring and insufficient information about the existing personnel system, this thesis presents only a prototype for implementation.

Chapter II discusses the Indonesian Navy personnel function activities and crucial problem in the present system, which the prototype is designed to address.

Chapter III reviews the database development process, which includes: the general system analysis and design, the database system concept and the advantages and disadvantages of using it, and introduction to database design (both logical and physical), with special emphasis on relational database design which focuses on the relational forms and the characteristics of relational databases.

Chapter IV describes the implementation of this personnel database system for Indonesian Naval Officers, particularly the Officer Career System.

Finally chapter V presents the conclusion that database processing applied to the Indonesian Naval Officer Career System, can increase decision making ease, timeliness, confidence, and reliability.

Programs using dBase II are provided so the user can easily interact with the computer to get information about the officer career field. These programs are shown in Appendix D.

II. BACKGROUND

A. INTRODUCTION

The Indonesian Navy has many subsystems, one of which is the Navy Personnel System. The Navy Personnel System is concerned with personnel resource utilization, and management of other resources to make the Indonesian Navy organizationally effective and efficient in supporting national defense and security.

B. PERSONNEL MANAGEMENT FUNCTION

The Indonesian Navy personnel system has two main functions; Manpower Management and Personnel Administration.

1. Manpower Management

In general the main functions of Manpower Management are: To investigate activities and to determine personnel requirements including qualification, quality and quantity for supporting the Indonesian Navy's main tasks.

2. Personnel Administration

Personnel activities have the following life-cycle functions: Procurement, Classification, Education and Training, Utilization, Treatment and Separation.

a. Procurement

Personnel procurement is the process of acquiring new personnel from national human resources according to Indonesian Navy Organization requirements.

b. Classification

Personnel classification is the process of analyzing and classifying individuals by job/occupation, so that individual requirements and organizational requirements are in balance.

c. Education and Training

Personnel education and training are the processes and activities which impact knowledge for personnel development related to assignments and promotions.

d. Utilization

Personnel utilization is the process and activity of selecting the right people for promotion and job/assignment. By placing the right person in the right place, the personnel assignment system facilitates maximum exposure and chances for individual promotion. Maximum exposure and promotion support future personnel assignments.

e. Treatment

Personnel treatment is a complex field, and is not an easy job. Activities in this category include salary, allowance (wife, children), morale, leave, recreation, rewards or awards, medical, insurance, etc.

f. Separation

Personnel separation is the process and activity of keeping a satisfactory balance within the organization while dealing with personnel retirement. Personnel retirement occurs:

- When manpower control and composition needs in the organization result in personnel separation (reductions in force).
- Because of normal retirement.

C. PERSONNEL SYSTEM REQUIREMENT

Information to support decisions by the Deputy of Personnel must be relevant, timely and correct.

The organization must maintain many files in the personnel system such as payroll, separation, reward, classification, education and occupation. In a file processing environment, these result in a high degree of data redundancy and are difficult to maintain and update. For example, after a promotion all files related to rank must be changed. Similarly, for personnel career planning, all education data, occupation data, and rank data must be provided on the same report.

It is not easy to retrieve information from different files in a short time, and it is inefficient to support these kinds of information needs using file systems. In order to avoid duplication, and to facilitate fast retrieval, a relational database system may be a desirable alternative.

Using a relational database system allows an organization's data to be processed as integrated files. It reduces artificially imposed constraints caused by separate applications and permits the Deputy of Personnel and his staff to access data more naturally.

D. THE CRUCIAL PROBLEM IN THE PRESENT SYSTEM

Personnel Administration cycles like training, career planning, promotions, job placement and separation pose many administrative problems. Training must have standardization grades,personal records for each trainee, personal biography etc. It is difficult to maintain and retrieve such data accurately and quickly.

The Navy Officer promotion system recognizes two periods in a year, April and October. So much data must be collected for supporting the candidate's promotion that it is difficult to retrieve data from the source. Sometimes human errors occur or too much data is collected.

When career planning is done by hand, the same problems recur because one must collect and gather data from a filing cabinet and then match it with each person and job.

When updating data related to promotion, a new job or new education it is difficult to manipulate data manually. When calculating a pension for example, it is difficult to provide information quickly because the birth date must be subtracted from the current year by manual methods. When the executive or the Chief of Staff needs adhoc information from the Deputy of Personnel, manual systems cannot provide results quickly, thus he cannot make the decision with confidence.

E. CAREER MANAGEMENT CONTROL

Three main activities occur in career control: promotion, education/training and job placement. Each closely supports the others.

1. Promotion

As mentioned earlier, the regular officer promotion system recognizes two periods in a year, April and October. Three or four months before, the candidates are proposed from naval stations/bases such as West Fleet, Marine Corps and other Main Commands. Much data must be collected to support the candidate's promotions, such as conduct, previous jobs, rank history, education and training history, and other administrative information.

Sometimes this information appears with manual typing error or incorrect information because of human errors, and much time is lost correcting it.

After collecting this information, the career control staff can examine and clarify which officers are eligible for promotion. Education, job placement and date of rank determine eligibility. The staff then matches this data with data from the personnel data bank. This procedure, if done by manual methods as is currently the case, is very time consuming and requires much staff and clerical support.

2. Educational

The educational system and its administration is a bit different from the promotion system. Information about education comes from centralized activities, but the same administrative process occurs as with the promotion system.

An example is the military education level from general development, for example the Second Officer Continuing Education corresponding to the US Army Officer Advanced Course. By collecting and gathering the data from filing cabinets, the career control staff can clarify and examine which officers are eligible to attend this course. After that they send a message/telegram to the Main Commands or Naval Stations. Main Commands then respond concerning the officer's elligibility for this course by sending a message back to Deputy for Personnel. Again the career control staff handles this information from all the Main Commands and Fleets. A lot of time is spent in clerical tasks, and even more staff is needed to handle educational activities.

3. Job placement

For regular job placement the peak load activities happen as the officer nears the end or finishes a course. After examining the officer's previous job experiences and educational background, the career control staff examines alternatives and decides where the officer would best serve in a new job. Again, if it is done by manual methods, sometimes information is incorrect and it is time consuming to correct. Just as with the promotion and educational systems, the officer data are kept in filing cabinets.

III. DATABASE DEVELOPMENT PROCESS

A. SYSTEMS DEVELOPMENT

The software engineering field is interesting and challenging because it requires the combination of both science and art. Software is an intangible thing, so it is more difficult to measure, evaluate and estimate than other engineering products.

1. The Roles of Systems Analyst

The Systems Analyst is known by various names such as Systems Engineer, Systems Designer, or Programmer Analyst. Currently System Analysts are usually located in or near the computer functions in an organization. It is most common to find them in a project development department which reports to the director of the information system, although Systems Analysis and Design are sometimes decentralized into the functional areas that process information.

A Systems Analyst tries to trace the current systems /manual systems and uses various means of information gathering techniques, such as interviews, questionaires, observation, and document examination. He must try to choose the techniques that best fit the situation of the user organization. He also must evaluate this information in order to recognize the problem and provide alternative solutions (Figure 3.1).

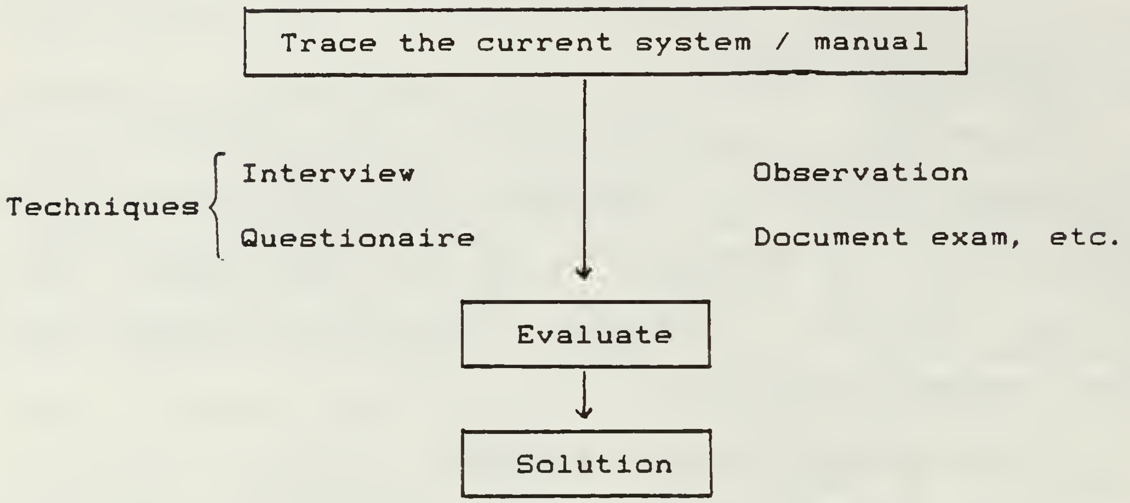
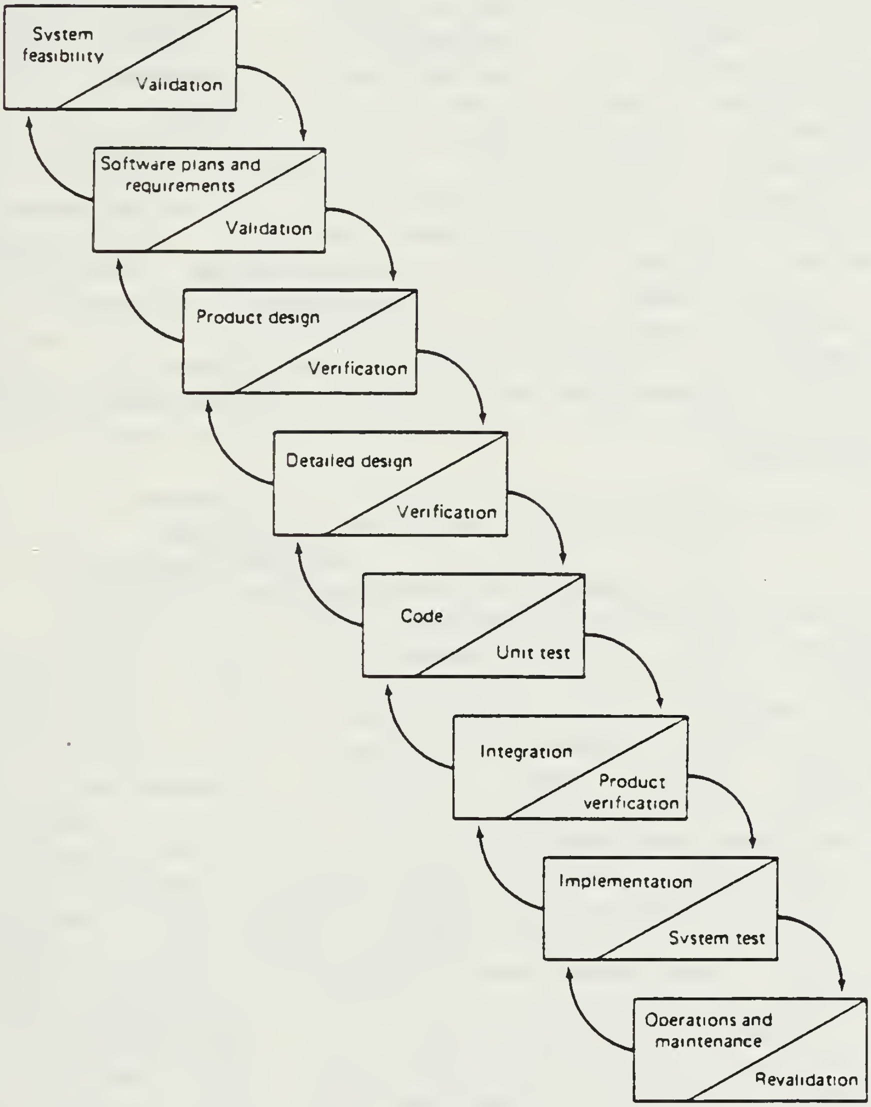


Figure 3.1 Tasks of Systems Analyst.

2. Software Life Cycle

The software engineering process is very complex. If software engineering is not provided with comprehensive and integrated tools, techniques and disciplines that support all activities of the development life cycle, productivity and quality can not be expected to realize substantial gains. The software life cycle may be fundamentally partitioned into the development phase and the operations and maintenance phase. There are many interpretations of the various development phases of software, but the important thing to remember is that the larger the project, the more detailed the engineering design phase must be. One methodology for software life cycle is the Waterfall model. [Ref. 2], is illustrated in figure 3.2. The original version was presented in Royce, 1970 and was foreshadowed in various U.S Air Force and industry publications such as Air Force, 1966 and Rosove, 1967.



(Reprinted from [Ref. 2])

Figure 3.2 Waterfall model.

As seen above, each phase culminates in a verification and validation activity. By verification and validation one can eliminate as many problems as possible in the results of that phase. So each product phase is the basis for the next product phase.

a. System Feasibility

System feasibility must consistently, comprehensively and accurately represent the user requirements. System feasibility is determining its life cycle feasibility and superiority to alternate concepts. The model of the system at the system feasibility level must support analysis and validation of the required operational capability as well as understanding the user and the developer.

b. Software Plan and Requirements

This phase determines the required functions, interfaces, and performance for the software product. Another activity is supporting the analysis of all subsystems from the system viewpoint such as communications and peripheral devices. The objective of this phase is to provide requirements for each of the subsystems. This involves trade-off studies of the overall system model to arrive at a complementary set of requirements for each subsystem that will, when integrated, collectively provide the behavior identified in the software plan and requirements.

c. Product Design

In this phase we consider the primary deliverable for software design. The design specification is a document with verified specification of the overall hardware/software architecture, control structure and data structure for the product. The design specification serves dual roles by providing a guide to software implementation (coding) and testing, and assisting the maintainer after software has been released. The specification can undergo considerable change during the software life cycle, therefore it is essential to review the design documentation at each step in the development phase. This phase can also provide a user manual's draft and test plan if necessary.

d. Detailed Design

With the use of a design representation that may be graphical, textual or tabular a detailed procedural specification for the software is created. Like a blueprint, the detailed design specification should provide sufficient information for someone other than the designer to develop the resultant source code.

e. Coding

The final objective of software engineering is to translate representations of software into a form that can be understood by the computer. In this phase the activity is a process that transforms a design into a programming language, such as Pascal, Cobol, Fortran, or dBase II

f. Integration

After each program component is tested individually, the next step is integration testing. The program components are put together and the application is tested in its entirety.

g. Implementation

In this phase the software product is fully operational on the hardware system.

h. Maintenance

Maintenance is critical in the database environment because changes to a data element can affect several application programs. During the use of a large program errors will occur which must be corrected.

There are many activities during maintenance such as:

1. Correction because of one or more coding errors.
2. Correction because of changing environment (new generation of hardware system, new operating system, new software version etc).
3. Correction to improve the software even though it may not have any errors.

B. DATABASE SYSTEM CONCEPT

Demands on data processing professionals to improve the effectiveness and efficiency of their organizations have caused them to turn to database technology. Unfortunately many organizations are switching to a database approach with little or no understanding of the concept.

Database systems in general stress controlled data redundancy, faster processing time, higher reliability, reduced storage requirements, program independence from changes in the storage structure, control of data administration, and a better definition of data. A database system is powerful, but to achieve these goals is not a simple matter.

1. Data as a Resource

Data has no value if it cannot be processed or retrieved in a timely and accurate manner. With the growth of computer capability in terms of more speed and memory, we can now process information rapidly and consistently.

Using file system applications, data is usually associated with a single functional application program, for example personnel data is associated with a personnel department program, logistics data is with a logistic department program, and payroll with a accounting department program. Some items might be stored in more than one functional area. For example, name and rank might be stored in the personnel department, accounting department and logistic department. Once the data itself is modified or changed, all the functional areas must change that data.

This is not efficient. As a result we get redundant and inconsistent data that increases the possibility of errors.

When using a database, the data must be commonly defined and consistently organized. Data must be organized so that they can be used by the entire system. Any update in one area must update all of the database records relating to this data regardless of where they are stored. This approach reduces and controls duplicate data storage and also makes integrated processing possible.

2. Database Management System

Database has been defined in many ways in the literature. Kroenke's definition [Ref. 3] is that a database is a collection of files and relationships among records in those files. Actually a database is more than a collection of files; it is a collection of integrated files.

A database management system (DBMS) is an integrated software package that allows users to define th e structure of the database, access the database and administer the database. One or more languages is provided to accomplish these tasks.

The data manipulation language (DML) is a language used to access the data in he database; a DML is used to perform insertions, deletions, changes, and retrievals. Also DML can be defined as an extension to a host programming language, such as COBOL, FORTRAN and query language (QL). The data definition language (DDL) is used to define the logical structure of the database and can be used to specify data items and their interrelationships.

The primary advantage of using a DBMS is that it provides data independence which means the database user does not need to be familiar with implementation details. Changes can be made to the data or to the schema with little impact on application programs using that data.

A second advantage is that it provides an integrated package of tools for handling data with greater standardization and control.

However, a general purpose DBMS might not be as efficient for a specific application as specialized software.

3. Database Administration

A database design project must have its own centralized administration. The systems development manager, therefore, must ensure that controls for the systems development project are not neglected.

Because of the shared nature of the data resource in the database approach, a centralized function is required to manage and protect it. Access to this data must be controlled. No function can be allowed to modify the data without the permission of the individual who is responsible for that. The Database Administrator is responsible to maintain access and security for all of the data.

Database Administration is a very complex job because the information which the user needs is varied and wide-ranging. Thus the Database Administrator must deal with providing the requested information. Issues to be considered include:

a. Whether different functional areas need to share the database.

b. Whether all levels of management (upper, middle, lower) need the information and how to structure the database to solve this problem.

c. The database structure which must be implemented to provide information in an accurate, relevant, and timely manner. The more information we need, the more data structure must be designed.

4. <u>Advantages and disadvantages of using a database</u>

File systems and database system each have their own advantages and disadvantages [Ref. 3: pp. 3-7].

a. Advantage of Using a Database:

1. Enables more information to be produced from a given amount of data.

2. Elimination of duplication saves file space, and to some extent, can reduce processing requirements. The most serious problem of data duplication is that it can lead to a lack of data integrity. A common result of a lack of data integrity is conflicting reports. Perhaps nothing is more aggravating to management or more embarrassing to the data processing staff than to be confronted with computer reports that disagree.

3. Creation of program / data independence.
Data independence means that when the data structure changes, application programs keep running without being changed.

4. Better data management.
When data are centralized in a database,

one department can specialize in the maintenance of data. That department can specify data standards and ensure that all data adhere to the standards. When someone has a data requirement, he or she can contact one department instead of many file maintenance groups.

5. Allows query language with easier programming and makes it easier to retrieve sophisticated information in DBMS environment.

b. Disavantages of using a database

1. Using a database can be expensive. The DBMS may occupy so much main memory that additional memory must be purchased. Conversion from existing systems can be costly, especially if new data must be acquired. Operating costs and overhead for some systems will be higher. For example, sequential processing of payroll will never done as fast in the database environment.

2. Database processing tends to be complex. Large amounts of data can be interrelated in the database. This means more sophisticated programming, and of course highly qualified systems and programming personnel are required.

3. Backup and recovery are more difficult. Because of increased complexity and because databases are often processed by several users concurrently, backup and recovery are more difficult.

4. Security more difficult. Integration, and hence centralization, increase vulnerability to security problems because all data are centralized under one system.

C. DATABASE DESIGN

A database contains a diversity of related data so that typically the equivalent several files are used to hold the data. Other files may be needed to hold descriptive information about the data and their relationships.

The analysis of databases concentrates on logical structures. The result of this analysis is a precise specification of the contents and required manipulations for the database.

An integrated collection of support programs and file structures to support a database, its logical requirements, and an interface to user program is called a database management system. For that reason alone, the design of a database will be the most important component affecting how well a system will work.

The design of a database describes how the data in the system is divided into the different files or records.

Database design basically has three steps:

- Decide what data should be stored.
- For the logical database design, identifying data entities, attributes, and relationships.
- For the physical database, it means physical storage of data on storage devices and the resultant performance of the database design.

1. <u>What Data Should be Stored</u>

If the system is at all complex,the designer needs to consider what kind of data should be stored. First he must look at the output or proposed output. Before the required output can be displayed on a screen or printed on a report, the data must be in the database somewhere. For example, if the system must print personnel addresses, it will need the name, street, city, zip code and so on. Or, if the system needs to know how old personnel were when they retired, the designer must store data from which age may be calculated (for example, birth date and the retirement year).

Most of the data entered will appear immediately on the output reports, but not always. The user may decide that a piece of data is not being used now, but might be needed in the future. The solution in such a case should be to enter it now. This may cause a little more data entry, but it is easier to have data entered as it is captured, rather than later when it may not be available.

2. Logical Database Design

A database contains many types of data items. These data items have to be associated into a database structure. A systems analyst tends to group together in records collections of data items which data processing views as being useful. The logical database structure is the foundation on which most future data processing will be built. Future organizations will be managed with database resources, networks to access the databases, and end-user software for employing and updating the data.

The logical design begins with an investigation of user requirements and ends with a logical description of a database that can support those requirements. Logical description such as data models present logical views of real entities and their relationships. An entity may be a tangible object such as a person. It may be intangible such as a job title. Entities have properties, called attributes, which associate a particular value from a domain of attribute values with each entity. Examples of attributes are color or name. Usually the domain for an attribute will be a set of integers, real numbers, or strings. An entity usually has a data item that uniquely identifies it.

For example, employee number is the unique identifier of the employee entity. Figure 3.3 is a sample of entities and attributes. Shown below the serial number is a unique key. Entity sets can be represented by a labeled box, see figure 3.4. Data models describes entity sets, entities, attributes and their relationships. Relationships among them are addressed by the relational model. Examples are relationships between college, departments, professors, and courses.

		Attributes	
	Serial #	Name	Address
	001234	David	Ocean Ave
Entities	002134	Mike	Seventh St
	006789	Fred	Third Ave

^
|
Entity #

Figure 3.3 Entity set example.

Figure 3.4 Entity set diagram.

College, department, professor and courses are identified as entity sets. Figure 3.5 shows the relationships between these entity sets. The relationship between college and its departments is called one-to-many, in that any one college may have many departments (Administrative, Computer, ORSA). The same relationship exists between departments and their courses.

Many-to-many is the relation between Professors and Courses: One professor can teach many courses, and one course can be taught by many professors.

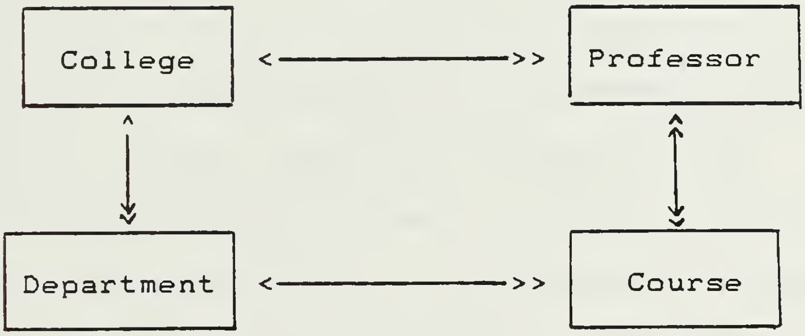


Figure 3.5 Relationship diagram.

3. Physical Database Design

The result of the physical database design is specification of the storage allocation parameters. Setting up the database on a real computer is called physical database design. In other words, the physical database is the component of the database that confirms the computerized version of the logical database.

The physical database consists of five components [Ref. 4: pp. 30-33]: storage structure, access strategy, data loading, data update, and data maintenance.

Knowledge of the storage structure is important because it indicates the power of the DBMS for certain applications. The storage structure is the physical architecture construction. Its consists of dictionary, indexes, relationships, data. The dictionary is the repository of the information represented by the compiled data definition language. Each DBMS's dictionary is unique.

For faster access, index mechanisms are needed. There are two kinds of indexes, primary and secondary. The primary indexes are unique value guaranteed and the secondary indexes are repeated values allowed.

The connections between record occurrences are called relationships. There are two kinds of relationships, static and dynamic.

The data part of he storage structure contains the occurrences of the records. Static DBMS have multiple record type data storage structures. Dynamic DBMS have a separate data storage structure for each record type.

An access strategy is a collection of routines that insert data into and retrieve data from the database. Understanding access strategy is basic to knowing why certain operations take longer than others.

Data loading is the process of inserting large quantities of data into the database at one time. Some DBMS allow or require data to be loaded in certain ways to achieve efficiencies during subsequent updating and reporting.

The updating process attempts to ensures current data. Critical in the updating application is how well the DBMS handles additions of new records, modification of element values, modifications of relationship occurrences and deletions of existing records.

The most important requirement for database maintenance is the creation of a backup copy of a database. While creating a backup is certainly the first step in backup-and-recovery, the nature of the backup process needs to be understood so that sufficient resources can be allocated to its accomplishment.

D. RELATIONAL DATABASE DESIGN

In a filing system, each functional area has it own files and procedures. Because of this there is a complex flow of paperwork between the functional areas. In this case data for different areas are separately maintained in separate files. It is hard to manage this data. Accuracy is lost, and maintenance and change are difficult to accomplish.

But when data are designed into an integrated database system, the data structures become more complex but the data flows are simplified. The data are consistent and accurate.

Changes in procedures can be made rapidly. Fundamentally different analysis and design techniques are needed.

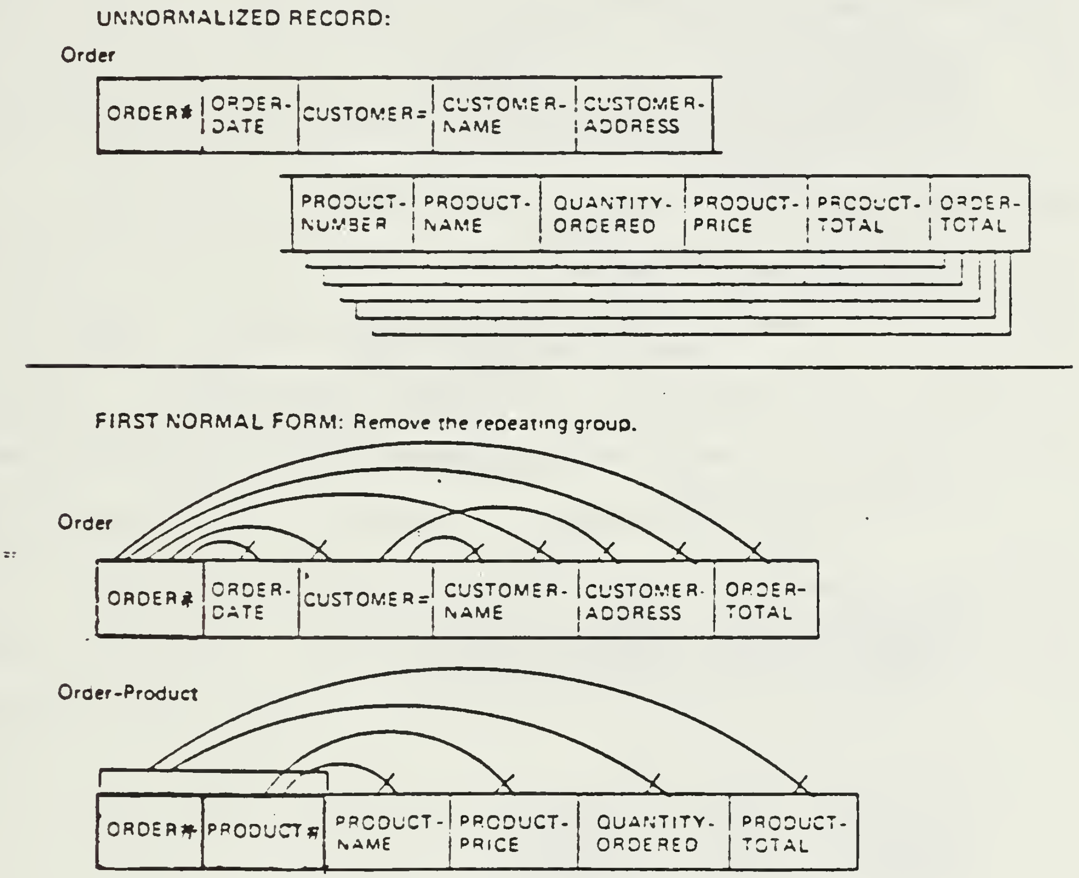
1. Relational normal forms

When modification of data has unexpected consequences, it is refered to as a modification anomaly. A deletion anomaly exists if the deletion of one entity deletes facts about a second entity. An insertion anomaly exists if an insertion of a fact about one entity cannot be made until a fact about a second entity is known. Anomalies can be removed or eliminated through normalization, but this normalization process can create an interrelation constraint.

2. Normal Forms

All relations are by definition in first normal form. When there are no repeating groups the relation is in first normal form. See figure 3.6

A relation is in second normal form if all nonkey attributes are dependent upon all of the key attributes. No attribute is dependent upon only part of the key. The second normal form is illustrated in Figure 3.7.



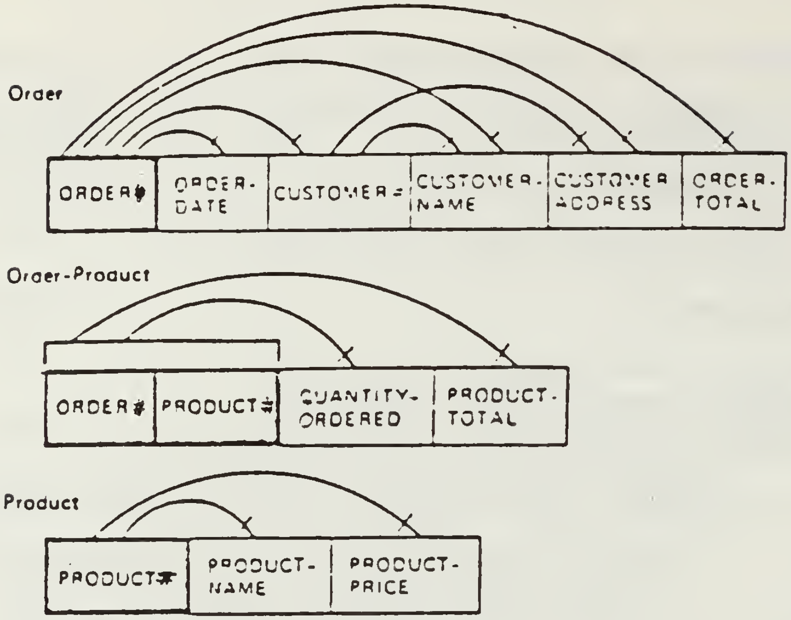
(Reprinted from [Ref. 5: p. 180])

Figure 3.6 From unnormalized to first normal form.

When relations are in second normal form, they may have transitive dependencies. These dependencies are removed by placing the relation in third normal form.

A relation is in third normal form if it is in second normal form and has no transitive dependencies. All attributes must depend upon all of the key, and dependencies are not transferred from one attribute to another. Third normal form is illustrated in figure 3.8.

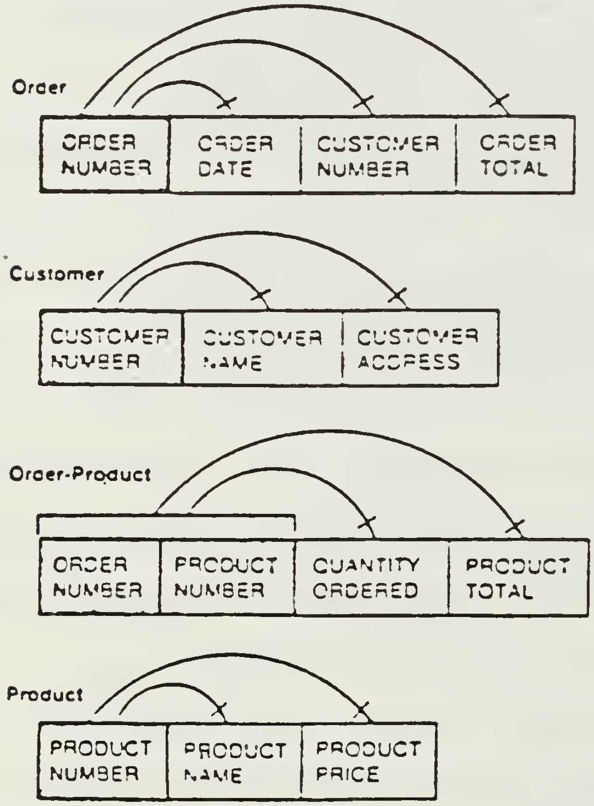
primary key as in the ORDER PRODUCT record above



(Reprinted from [Ref. 5: p. 181])

Figure 3.7 Second Normal Form.

THIRD NORMAL FORM: Remove attributes dependent on data item(s) other than the primary key, as in the ORDER record above.



(Reprinted from [Ref. 5: p. 181])

Figure 3.8 Third Normal Form.

3. Relational Database Design Criteria

The first design criterion is elimination of modification anomalies. If relations cannot be put into normal form, then modification anomalies do not happen. Relations that are in normal form are prefered, and normal for that is become is a design objective.

Relation independence is the second design criterion. Two relations are independent if a modification can be made to one relation without affecting the other relation. Relation dependence is sometimes caused by normalization to remove modification anomalies. This effect causes a conflict in design criteria.

The third criterion is ease of use. The design should structure relations so that they are familiar and natural to the users.

IV. DATABASE IMPLEMENTATION

A. INTRODUCTION

The main purpose of this chapter is to design a database system which may be used by the Indonesian Navy Deputy of Personnel to make more effective decisions.

Personnel management was previously defined as those managerial activities that deal with procurement, development, maintenance, utilization and separation. This is also known as the personnel management cycle. For supporting these activities, data must be accurate and timely. All of the personnel activities are controlled by the Deputy of Personnel. These include education, training, promotion, and job evaluation.

The computer can help support the Deputy of Personnel with various decision making processes because of its speed and accuracy in processing large amounts of data.

An organization can more easily decentralize its operations when it uses interactive processing. This adds a new dimension to the Personnel Management Information System by allowing easy editing and filing of existing and new data.

After careful analysis, it was decided that the functional requirement described in Chapter II suggested the application of a relational database. This justification was based not only upon various possible record relationships, but also the frequency and volume of the expected transactions. Promotions occur regularly twice a year. Officers are selected for promotion based on requirements because of job vacancies particularly for the colonel and flag officer. For this purpose the system should be designed with a query capability.

For example the amount of officer records is around 8000, with five to seven percent eligible for promotion, and because of limited vacancies only two or three percent actually are promoted accepted.

The same thing occurs in educational and job placement. Comparing the officers' chances for various schools, the most significant hurdle is from Second Officer Educational level to Command and Staff College Level. This happens once a fiscal year and is usually followed by job placement. This kind of job placement is predictable, so it can be prepared as the officer nears the end or finishes a course. But managers still need some queries to handle this problem, because there are many irregular job placements because of job requirements from outside the Navy organization, for example D O D. The system should be able to provide relevant information immediately to support fast decision making by the Director of Personnel or Staff to fill those job vacancies.

The above is based on career applications only. Besides this, there are other applications such as rewards and pension. Again, the system should eventually be able to handle all personnel officer activities.

B. RELATIONAL DATABASE DESIGN AND APPLICATION

There are many ways in which a database can be designed. This design will describe a design theory and application for the Indonesian Naval Officer Personnel System, particularly the career system, where education, rank and job evaluation are involved.

The most important phase in the design of a database is in the relational database design, where the record relationship and record structure will be developed.

1. Record_Relationship

Relationships can be specified in a variety of ways. One technique is called a Bachman diagram or structure diagram. See figure 4.1.

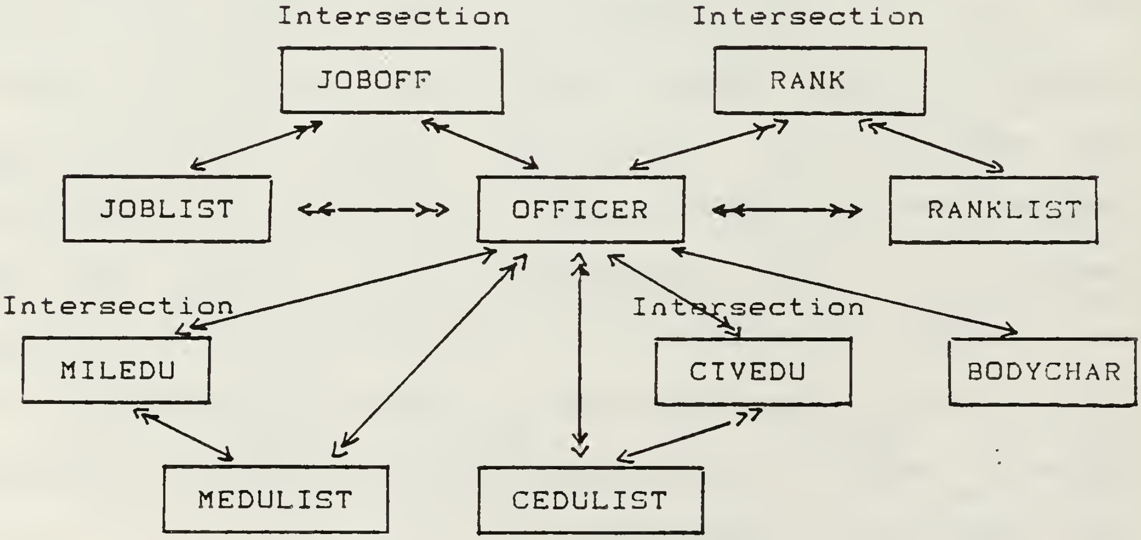


Figure 4.1 Bachman diagram

The above diagram shows the relationships among records. The single and double arrow notation is used to express relationships among records. There are four different relationships, one-to-one, one-to-many, many-to-one, and many-to-many relationships.

Examples as follows:

a. One-to-one.
An officer can only have one set of body characteristics and vice versa.
OFFICER <-----> BODYCHAR.

b. One-to-many.
One officer may have attended several military schools.
OFFICER <----->> MILEDU.

c. Many-to-one.
Many differents jobs in job records history will have been performed by one officer.
JOBOFF <<-----> OFFICER.

d. Many-to-many.
An officer can be eligible for many jobs or vice versa.
OFFICER <<----->> JOBLIST.

2. Record Structure

Record structure shows the relationships among attributes, where the key attribute is underlined and uniquely identifies each record. For many-to-many relationships between records, another record is needed, called an intersection record. This record is device for integrating two or more files and the keys of the files being integrated must appear in the intersection record. Examples of record structure as follows:

1. Personal identification
OFFICER (SERNO, NAME, CORPS, RANKID, EDUID, SEX, BIRTHDD, BIRTHMM, BIRTHYY, BIRTHP, RELIGION)

2. Body characteristic
BODYCHAR (SERNO, WEIGHT, HEIGHT, SKCOLOR, HRTYPE, EYCOLOR, BLOOD, SHOE, HAT, UNIFORM)

3. Rank (intersection)
RANK (SERNO, RANKID, RANKDD, RANKMM, RANKYY, RORNO, RDATORNO)

4. Military education (intersection)
MILEDU (SERNO, MEDUIRANK, MEDLEVID, MEDUID, MDDEDST, MMMEDST, MYYEDST, MDDEDED, MMMEDED, MYYEDED, MEDUDUR, MEDURSLT)

5. General education (intersection)
CIVEDU (SERNO, CEDURANK, CEDLEVID, CEDUID CDDEDST, CMMEDST, CYYEDST, CDDEDED, CMMEDED, CYYEDED, CEDUDUR, CEDURSLT)

6. Occupation (intersection)
JOBOFF (SERNO, JOBID, JOBRANK, JOBDD, JOBMM, JOBYY, JOBECH, JORNO, JDATORNO)

7. Occupation list
JOBLIST (JOBID, JOBDES, ECHELON, JOBSTAT)

8. Rank requirement for occupation (intersection).
JOBRNK (JOBID, JOBRK)

9. Corps requirement for occupation (intersection).
JOBCORP (JOBID, JOBCO)

10. Military education requirement for occupation (intersection). JOBEDU (JOBID, JOBED)

11. Military education list
MEDULIST (MEDUID, MEDUDES)

12. General education list
CEDULIST (CEDUID, CEDUDES)

13. Rank list
RANKLIST (RANKID, RANKDES, RKECHREQ)

C. INTERRELATIONAL CONSTRAINTS

As Kroenke says [Ref. 3: pp. 286, 305], not all relational database designs are equal, some are better than others. The important point is, a design that meets the user's needs is better than one that does not.

When a relation is normalized, this often may eliminate modification anomalies at the expense of creating interrelational constraints. The elimination of anomalies may, however have a disadvantage as well. Everytime a record is split into two or more relations an interrelational constraint is created. The initial job relation had modification anomalies because it was not in 2nd normal form. We eliminated these anomalies by splitting the relation into the "JOBOFF" and "JOBLIST" relations.

The interrelation constraints occur, because two relations share the attribute "JOBID". However a "JOBID" may not exist in "JOBOFF" if it is not in "JOBLIST".

We can say that the values of "JOBID" in "JOBOFF" must be a subset of the values on "JOBID" in "JOBLIST". See figure 4.2.

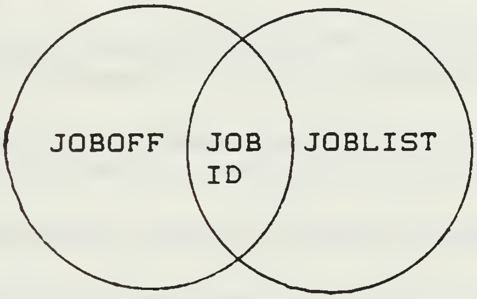


Figure 4.2 Example of JOB relational.

D. THE DESIGN APPROACH

The design of this system was approached by considering the output required and anticipating the questions and queries that management might ask. Output would be provided in both screen and printed form.

The following are examples of some queries and output form:

1. Query rank.

   a. Gives the information about all of officers in a certain rank, screen or printer.

   b. List an officer's promotion history by entering his serial number, screen or printer.

2. Query education.

   a. Gives the information about all officers who have a military education level (e.g. Command and Staff College level), screen or printer.

   b. Gives the information about all o fficer who have a specified military education (e.g. Navy Command and Staff College), screen or printer.

3. Query job.

   a. List all officers eligible for a certain job, screen or printer.

   b. List all jobs which an officer may be eligible for, screen or printer.

4. Print personnel.

   a. Gives all information about all of the officers list by by name (screen or printer).

   b. Gives all information about all of the officers listed by serial number (screen or printer).

Based upon the estimated outputs, the next step was to define what kind of data was required to support these outputs and what kind of record structure should be included into the dBase II database. (See data dictionary Appendix B.)

E. SYSTEM CONSTRAINTS

The system has certain hardware and software constraints. These constraints are :

1. This system must use DOS operating system version 2.0 or higher.

2. This system uses dBase II database management program.

3. This system runs on IBM PC or compatible only.

4. The widths of the output are less than 80 columns.

5. In order to make additions, changes and deletions, the users must be familiar with the design of the system and know about this data structure.

F. SYSTEM EXPANDABILITY

By the design of its data structure, this system can be expanded. Some of the possible expansions are:

1. The system can be expanded by creating and adding data files such as:

   a. Personal identification data structure (birth date, birth place).

   b. Personal characteristics (weight, height, hair color, eye color, etc).

   c. Family data (spouse, children, address, sex, birth date, and birth place).

   d. Payroll (main salary, wife allowance, children allowance, and deductions).

   e. Uniform sizes and other possible personal data.

2. By adding the above to the data structure, the output can be easily expanded to include:

   a. Output reports for payroll.

   b. Output reports for personal characteristics.

c. Output reports for occupation.

d. Output reports for families, etc.

3. Updating modules for the addition of data files as described above.

G. HIERARCHY CHART OF THE SYSTEM

The diagram below (figure 4.3) is the hierarchy chart of the system, where each block/module has a specific function, such as adding or updating a record. Each module has submodules with similar functions but different tasks, such as officer record addition, job record addition, etc.

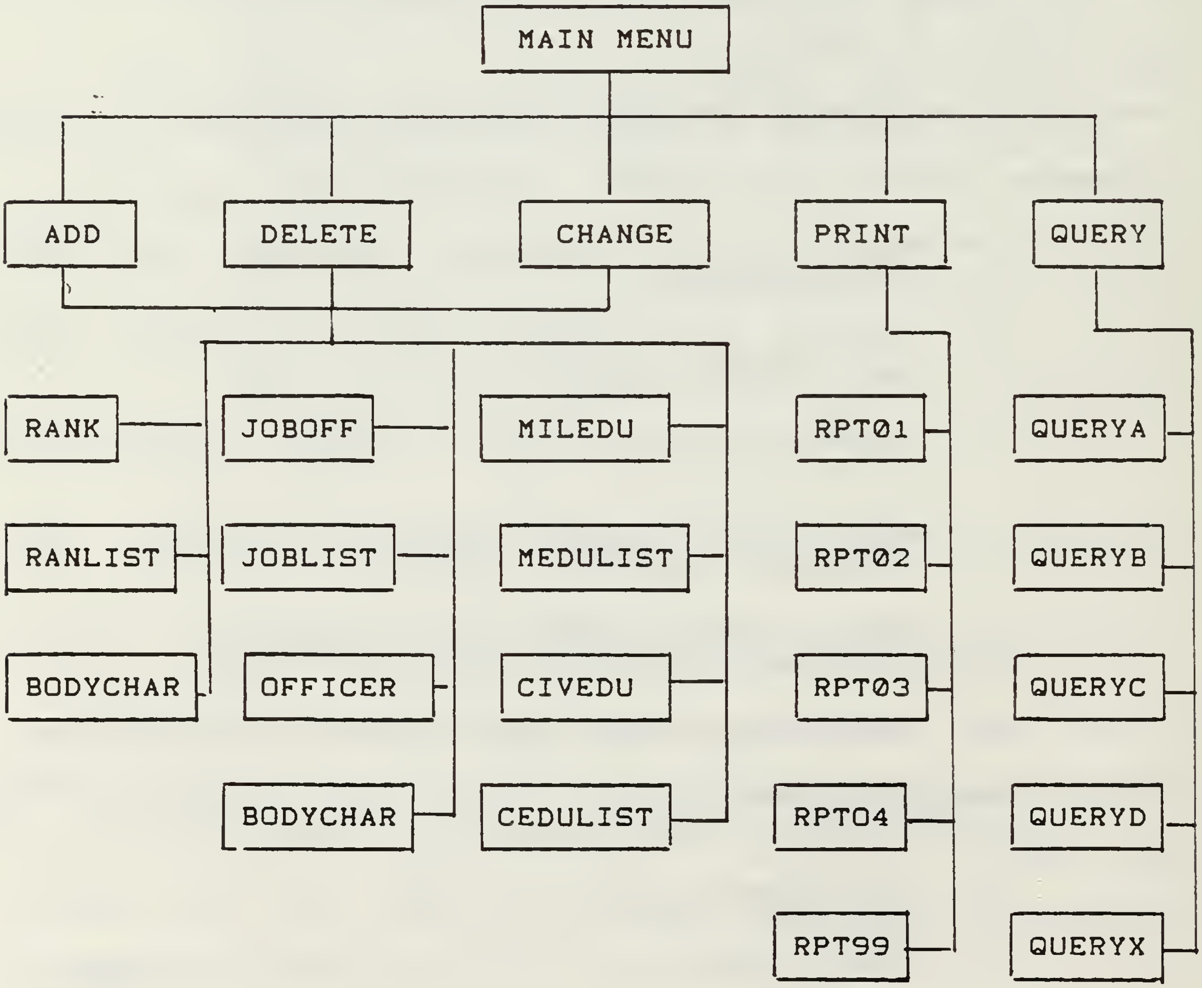


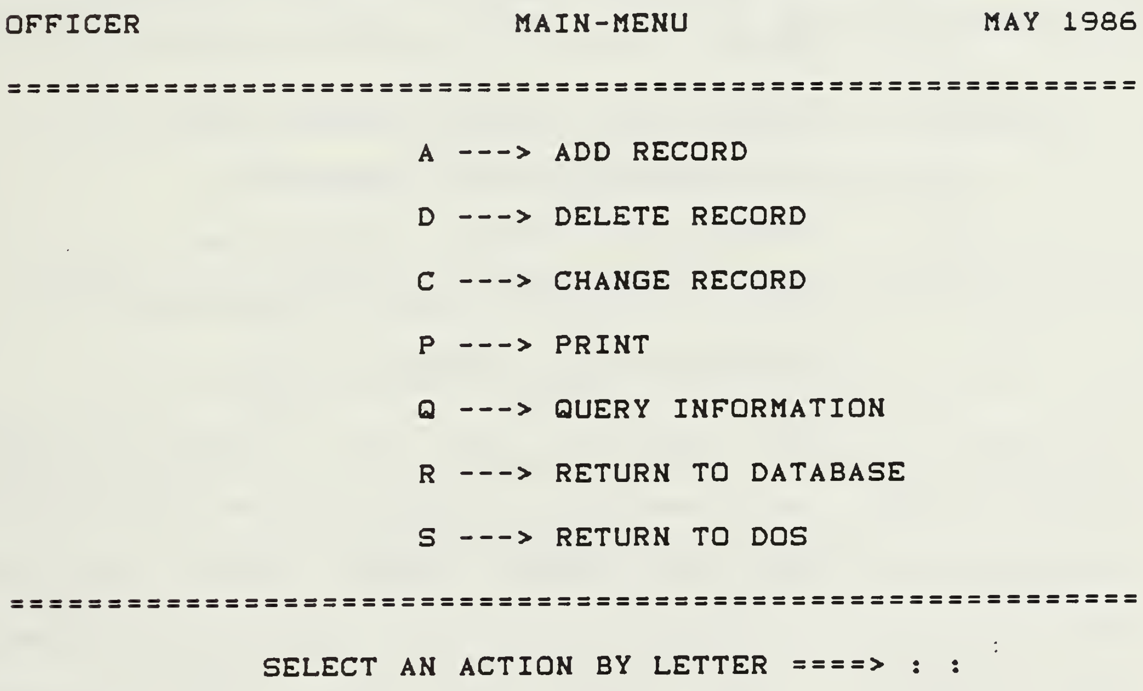
Figure 4.3 Hierarchy chart of the system.

H. IMPLEMENTATION

As mentioned before, the design of this system was approached by considering and anticipating the questions and queries that management might ask. The following are **examples** of queries relating to educational and job information.

1. Educational Information

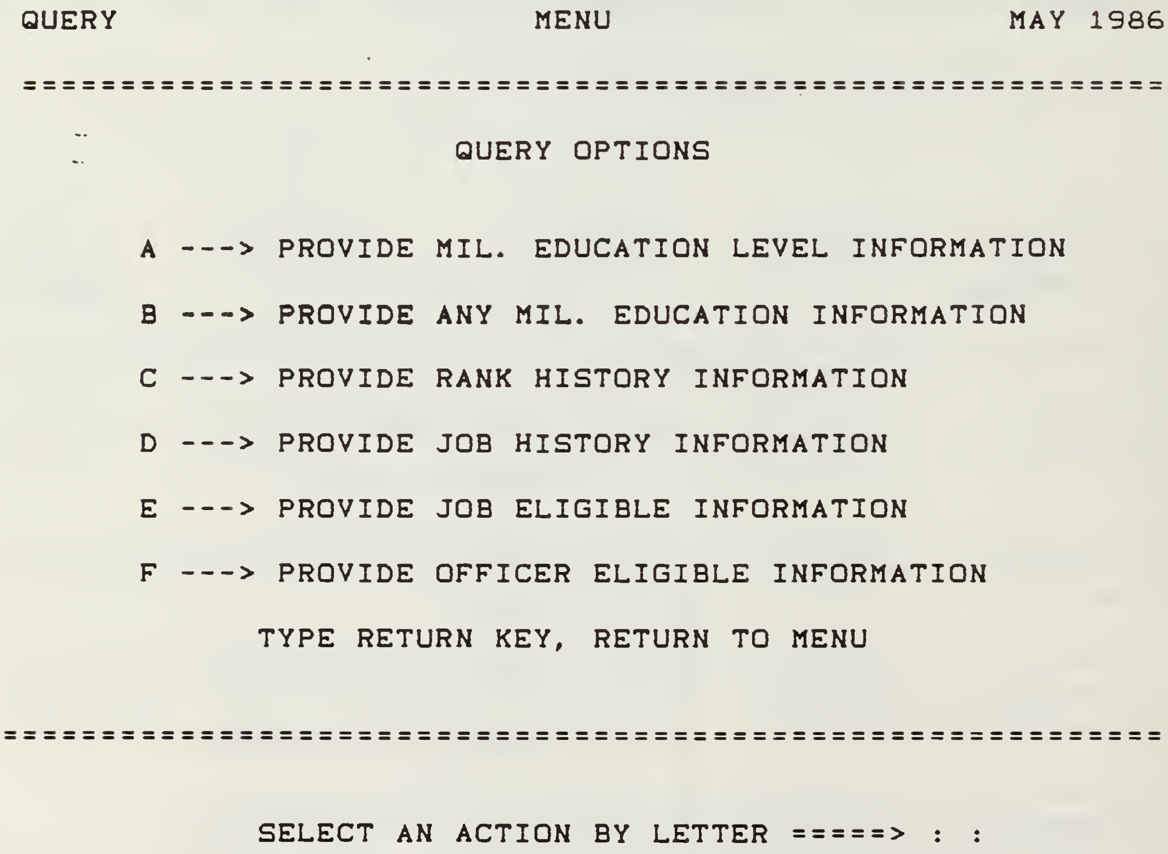
After successfully "booting" the system and reaching the main menu by following the instructions in the user manual (Appendix A), the user can choose one of the main menu selections (igure 4.4).



```
OFFICER                      MAIN-MENU                     MAY 1986
=======================================================================

                        A ---> ADD RECORD

                        D ---> DELETE RECORD

                        C ---> CHANGE RECORD

                        P ---> PRINT

                        Q ---> QUERY INFORMATION

                        R ---> RETURN TO DATABASE

                        S ---> RETURN TO DOS

=======================================================================
              SELECT AN ACTION BY LETTER ====> : :
```

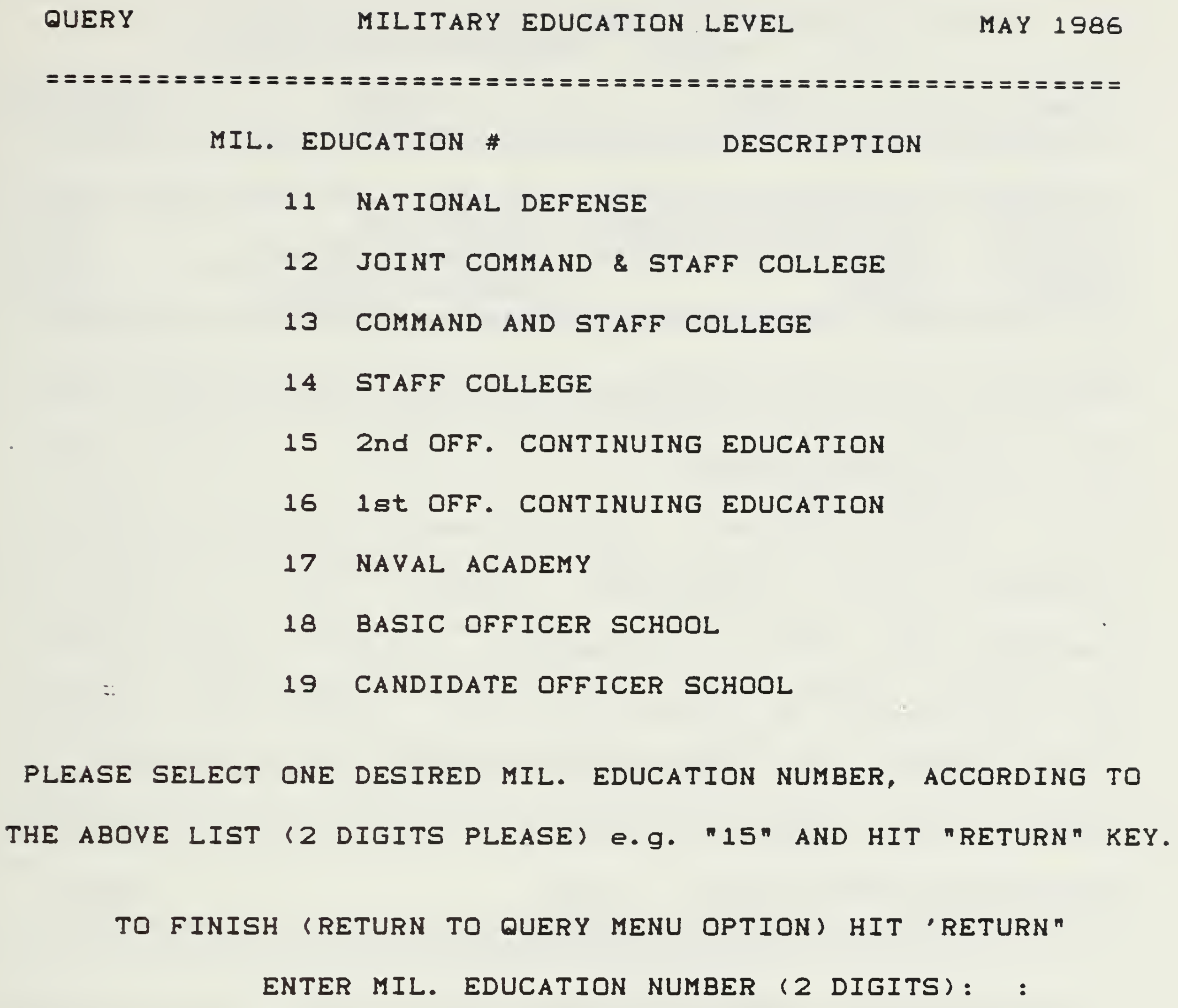
Figure 4.4 Main menu selections.

Selecting the <Q> option from the menu result in a query menu selection (see Figure 4.5) and again the user chooses one of them. For selecting <A> will provide the military education level for an officer. The next screen show the list of military education level and the corresponding code (see Figure 4.6).



```
QUERY                         MENU                        MAY 1986
==================================================================

                          QUERY OPTIONS

      A ---> PROVIDE MIL. EDUCATION LEVEL INFORMATION
      B ---> PROVIDE ANY MIL. EDUCATION INFORMATION
      C ---> PROVIDE RANK HISTORY INFORMATION
      D ---> PROVIDE JOB HISTORY INFORMATION
      E ---> PROVIDE JOB ELIGIBLE INFORMATION
      F ---> PROVIDE OFFICER ELIGIBLE INFORMATION
             TYPE RETURN KEY, RETURN TO MENU

===================================================================

             SELECT AN ACTION BY LETTER =====> : :
```

Figure 4.5 Query Menu Selection.



```
QUERY              MILITARY EDUCATION LEVEL           MAY 1986
==================================================================
          MIL. EDUCATION #             DESCRIPTION

               11  NATIONAL DEFENSE

               12  JOINT COMMAND & STAFF COLLEGE

               13  COMMAND AND STAFF COLLEGE

               14  STAFF COLLEGE

               15  2nd OFF. CONTINUING EDUCATION

               16  1st OFF. CONTINUING EDUCATION

               17  NAVAL ACADEMY

               18  BASIC OFFICER SCHOOL

               19  CANDIDATE OFFICER SCHOOL

 PLEASE SELECT ONE DESIRED MIL. EDUCATION NUMBER, ACCORDING TO
THE ABOVE LIST (2 DIGITS PLEASE) e.g. "15" AND HIT "RETURN" KEY.

      TO FINISH (RETURN TO QUERY MENU OPTION) HIT 'RETURN"
             ENTER MIL. EDUCATION NUMBER (2 DIGITS):  :
```

Figure 4.6 Example of Menu for Military Education Level.

By typing <13> in response to the menu in Figure 4.6 (code for Command and Staff College level), a list of officers currently attending the Command and Staff College including serial number, name, corps and the year end's attended (see Figure 4.7) will be displayed. Before the list appears on the srceen, the system asks if the user wants the display on the screen or in hardcopy form.

```
                  EQUIVALENT WITH THE LEVEL OF
                  2nd OFF. CONTINUING EDUCATION

 NAME            RANK                  SCHOOL              YEAR-END
=====================================================================

BUDIYONO  LIEUTENANT COLONEL  2nd OFF. CONTINUING EDU.  1983
MARTIN    LIEUTENANT COLONEL  A.G ADVANCE               1983
SUWITO    LIEUTENANT COLONEL  INFANTRY ADVANCE          1983
SULIS     MAJOR               INFANTRY ADVANCE          1983
=====================================================================
```

Figure 4.7 Example of Officers Attending Military Edu. Level

2. Job_Information

The procedure to retrieve educational information is similar. The job information system tries to match the desires of an officer concerning eligible jobs placement and conversely a certain job for which many officers are eligible. In this prototype system a simple example is given to show a personnel officer trying to match an officer with a job description.

Each job needs eligible officers to fill it. In a simple case the basic prerequisites for job placement are rank, education and corps. Secondary considerations include other adjustment and past performance.

Given a job description and prerequisites, this prototype can help the user to match a certain job with the officers eligible by reason of rank, education, and corps consideration. Exact job descriptions and eligibility requirements are beyond the scope of this thesis. We are only suggesting how this system might expedite the job placement matching process.

After initial screening it is a simple matter for the system to output an officer's promotion history and job history to make complete information available. An example follows:

Start with the main menu and respond with the appropriate query selection; then choose or type <E> for jobs eligible for an officer (see Figure 4.8). By giving the input officer's serial number, the screen will appear with two selections: job available for the same rank, or after promotion (see Figure 4.9). After typing <s> the system responds by asking if the user needs the list on the screen or hardcopy. By typing <h> the computer gives all information on job placement after promotion correlated by rank, education, and corps (see Figure 4.10).

```
QUERY                          MENU                          MAY 1986
=====================================================================
                          QUERY OPTIONS

      A ---> PROVIDE MIL. EDUCATION LEVEL INFORMATION

      B ---> PROVIDE ANY MIL. EDUCATION INFORMATION

      C ---> PROVIDE RANK HISTORY INFORMATION

      D ---> PROVIDE JOB HISTORY INFORMATION

      E ---> PROVIDE JOB ELIGIBLE INFORMATION

      F ---> PROVIDE OFFICER ELIGIBLE INFORMATION

               TYPE RETURN KEY, RETURN TO MENU
=======================================================================
               SELECT AN ACTION BY LETTER =====> : :
```

Figure 4.8 Menu for Job Selection.

```
     JOB AVAILABLE FOR THE SAME RANK or AFTER PROMOTION ?

           ENTER 's' FOR SAME RANK or 'p' PROMOTION
```

Figure 4.9 The option for same rank or after promotion.

```
                          JOB ELIGIBLE FOR :

003333    SUWITO                LIEUTENANT COLONEL      MARINES

JOB-ID          DESCRIPTION              STATION              CITY
==============================================================

 003   ASS. LOGISTIC                 MARINE H.Q         JAKARTA
 008   ASS. PERSONNEL                D O D              JAKARTA
 012   INSTRUCTOR                    NAVAL ACADEMY      SURABAYA
==============================================================
```

Figure 4.10 Example of jobs placement for an officer.

Conversely, given a job-id, the system will respond with all eligible officers with the same consideration given to education, rank and corps.

3. Special Job Discussion

The system described above is just a prototype, and further special work is needed. Job placement is a critical task because many factors must be considered, such as education, rank and corps.

If possible, the needs of the organization should be synchronized with the individual's needs and desires, in order to make the officers more satisfied with their jobs/ tasks.

There is a further requirement for definitions of job descriptions and eligibility requirements. This is not an easy task because there are hundreds of jobs which must be carefully analyzed. More research is needed to accomplish this task before the system can be expanded.

Because of the very complex consideration involved in personnel management, this system remains just a tool to get more reliable, up to date and timely information to the user.

V. CONCLUSION AND RECOMMENDATION

The Naval Officers Personnel System is very complex. Managing it manually demands great effort, is time consuming, requires a large staff to manage the personnel function, and is neither effective nor efficient in supporting decision making processes. Database processing can increase decision making productivity and provide relevant, accurate, and timely information.

This thesis has focused on a proposed Personnel Database System for Indonesian Naval Officers. However this system can be used for other personnel applications with a little modification. The developed sample database presented here is based on relational normal form. Normal forms can be applied to decrease inefficiency of the relational database model in the system design process, and also to decrease data redundancy.

A sample implementation using dBase II is provided in Appendix A, showing user interaction with the computer without necessarily knowing programming or database system management.

With the latest microcomputer performance and capabilities in networking, this system could be put on staff personnel desks and connected to each other in one headquarters building via a distributed network system, without requiring more space or restructuring office setting.

A complete implementation of the system needs more data about job descriptions and prerequisites regarding education, rank, corps, sex and other considerations. Finally the system must evolve through continued use and feedback from personnel management users.

APPENDIX A

USER MANUAL

1. PURPOSE

The purpose of this manual is to describe how to run the system.

2. OBJECTIVES

This system was designed to be user friendly, expandable and maintainable.

3. CONFIGURATION OF THE SYSTEM

To run this system the following is necessary :

a. IBM PC or compatible.

c. IBM printer or compatible.

d. 2 disk drives

e. dBASE II diskette.

f. dBASE II Project diskette.

4. RECORD STRUCTURE.

There are nine record types and fifteen table codes within this system. The records and table codes are:

a. Officer identification record (OFFICER.DBF)

b. Rank record (RANK.DBF)

c. Military education record (MILEDU.DBF)

d. General education record (CIVEDU.DBF)

e. Occupation record (JOBOFF.DBF)

f. Occupation list (JOBLIST.DBF)

g. Body characteristic record (BODYCHAR.DBF)

h. Corps code table (CORPCODE.DBF)

i. Rank list (RANKLIST.DBF)

j. Military education list (MEDULIST.DBF)

k. General education list (CEDULIST.DBF)

l. Sex code table (SEXCODE.DBF)

m. Religion code table (RELIGION.DBF)

n. City code table (CITYCODE.DBF)

o. Education result table (RESULT.DBF)

p. Skin color table (COLORSK.DBF)

q. Hair color table (TYPEHR.DBF)

r. Eye color table (COLOREYE.DBF)

s. Blood table (BLOOD.DBF)

t. Uniform size table (UNIFORM.DBF)

u. Echelon code table (ECHELON.DBF)

v. Rank requirement for occupation (JOBRNK.DBF)

w. Education requirement for occupation (JOBEDU.DBF)

x. Corps requirement for occupation (JOBCORP.DBF)

For a ore detailed description of the data records and tables, see Appendixes B.

5. RUNNING THE SYSTEM

a. Turn on the following devices, in order:

1. IBM PC/compatible system unit (switch on the right side).

2. The monitor (switch located on the front side).

3. The printer (Okidata, switch located on the right side).

b. Insert Operating system diskette into drive A, and the system into drive B.

c. Wait for the system to boot up, answer the prompts by entering date and time or just hit <Enter> twice.

d. When the screen format appears type <B:> to move drive B.

e. Now type <dbase pers>

f. When the system asks "enter your password please", simply type <pers>; If you fail in this section because you type the wrong password (unauthorized access), then type <do pers>.

g. Now you are in the OFFICER system which displays the main menu. The main menu will give you the choice of the following options

```
A ---> ADD RECORD

D ---> DELETE RECORD

C ---> CHANGE RECORD

P ---> PRINT

Q ---> QUERY INFORMATION

R ---> RETURN TO DATABASE

S ---> RETURN TO DOS
```

Once you are at the main menu level, just type the desired option (A,D,C,P,Q,R,S).

APPENDIX B

DATA DICTIONARY

This data dictionary contains data elements of the database of the personnel system. There are 5 columns in the table:

1. Data item. This column contains the data item as it is known to the user.

2. Acronym. This column contains the unique name for the data item that will be used by programmers/ analysts.

3. Type. This column contains the data item's type. "N" means numeric and "C" means character (alphabetic).

4. Length. This column contains the number of characters used in each data item.

5. Description. This column contains the description of the data item.

1. Officer identification OFFICER

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Serial #	SERNO	C	6	serial number, primary key
Name	NAME	C	25	The name of personnel
Corps	CORPS	C	2	Corps code (see CORPCODE table 01)
Present rank	RANKID	N	2	Present rank (see RANKLIST table 02)
Education	EDUID	N	2	Last military education (not available)
Sex	SEX	N	1	Sex code (see SEXCODE table 05)
Birth date	BIRTHDD	N	2	DD (Day)
Birth month	BIRTHMM	N	2	MM (Month)
Birth year	BIRTHYY	N	2	YY (Year)
Birth place	BIRTHP	C	20	City
Religion	RELIGION	C	1	See RELIGION table 06

2. Body characteristic BODYCHAR

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Serial #	SERNO	C	6	serial number, primary key
Weight	WEIGHT	N	3	Weight in kg
Height	HEIGHT	N	3	Height in cm
Skin color	SKCOLOR	N	1	Color of skin (see COLORSK table 09)
Hair type	HRTYPE	N	1	Type of hair (see TYPEHR table 10)
Eye color	EYCOLOR	N	1	Color of eyes (see COLOREYE table 11)
Blood type	BLOOD	N	1	Type of blood (see BLOOD table 12)
Size of shoes	SHOE	N	2	In cm
Size of hats	HAT	N	2	In cm
Size of uniform	UNIFORM	N	1	3 sizes (see UNIFORM table 13)

3. Rank RANK (Intersection)

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Serial #	SERNO	C	6	Serial number, primary key
Rank ID	RANKID	C	2	Rank code (see RANKLIST table 02)
Rank date	RANKDD	C	2	Date of rank
Rank month	RANKMM	C	2	Month of rank
Rank Year	RANKYY	C	4	Year of rank
Order Number	RORNO	C	8	Order Number with format Number(4)/Month(2)/Year(2)
Date Number	RDATORNO	N	6	DDMMYY

4. Military Education MILEDU (Intersection)

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Serial #	SERNO	C	6	Serial number, primary key
Rank	MEDURANK	N	2	Rank in education (see RANKLIST table 02)
Education level	MEDLEVID	C	2	Level of education (see MEDLEV table 03)
Education ID	MEDUID	C	2	Military education (not available)
Date start	MDDEDST	N	2	Date education start
Month start	MMMEDST	N	2	Month education start
Year start	MYYEDST	N	2	Year education start
Date end	MDDEDED	N	2	Date education end
Month end	MMMEDED	N	2	Month education end
Year end	MYYEDED	N	2	Year education end
Duration of education	MEDUDUR	N	3	Duration of education (Weeks)
Place of education	MEDUPLCE	C	3	Place of education (City) (see CITYCODE table 07)
Educ. Result	MEDURSLT	N	1	Result of education (see RESULT table 08)

5. General Education CIVEDU (Intersection)

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Serial #	SERNO	C	6	Serial number, primary key
RANK	CEDURANK	N	2	Rank in education (see RANKLIST table 02
Education Level	CEDLEVID	C	2	Educational level (see CEDLEV table 04)
Education ID	CEDUID	C	2	General education (not available)
Date start	CDDEDST	N	2	Date education start
Month start	CMMEDST	N	2	Month education start
Year start	CYYEDST	N	2	Year education start
Date end	CDDEDED	N	2	Date education end
Month end	CMMEDED	N	2	Month education end
Year end	CYYEDED	N	2	Year education end
Duration of education	CEDUDUR	N	3	Duration of education (Weeks)
Place education	CEDUPLCE	C	3	Place of education (see CITYCODE table 07)
Educ. Result	CEDURSLT	N	1	Result of education (see RESULT table 08)

6. Occupation JOBOFF (Intersection)

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Serial #	SERNO	C	6	Serial number, primary key
Job ID	JOBID	C	3	Job code (see JOBLIST table 15)
Rank	JOBRANK	N	2	Job rank (see RANKLIST table 02)
Job date	JOBDD	C	2	Date of job
Job month	JOBMM	C	2	Month Job
Job Year	JOBYY	C	4	Year Job
Job Echelon	JOBECH	N	2	Job of echelon (see ECHELON table 14)
Order Number	JORNO	C	8	Order Number with format Number (4)/Month(2)/Year(2)
Date Number	JDATORNO	N	6	DDMMYY

7. Rank requirement for occupation JOBRNK (Intersection)

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Job ID	JOBID	C	3	Job code (see JOBLIST table 15)
Job rank	JOBRK	C	2	Rank requirement (see RANKLIST table 02)

8. Education requirement for occupation JOBEDU (Intersection)

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Job ID	JOBID	C	3	Job code (see JOBLIST table 15)
Education req.	JOBED	C	2	Mil. education requirement (not available)

9. Corps requirement for occupation JOBCORP (Intersection)

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Job ID	JOBID	C	3	Job code (see JOBLIST table 15)
Job corp	JOBCO	C	2	Corp requirement (see CORPCODE table 01)

APPENDIX C

SAMPLES OF TABLE CODE

01. Table of Corps Code CORPCODE

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Corps ID	CORPS	C	2	Corps code
Corps desc	CORPSDES	C	10	Corps name

01	Line
02	Engineer
03	Electronics
04	Marines
05	Administration
06	Specialist
07	Health
08	Woman

02. Table of Ranks RANKLIST

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Rank ID	RANKID	C	2	Rank code
Description	RANKDES	C	20	Rank title
Education requirement	RKEDUREQ	C	2	Education requirement (see RNKEDURQ table)
Echelon requirement	RKECHREQ	C	2	Echelon requirement (see RNKECHRQ table)

11	Admiral / General
12	Vice Admiral / Lieutenant General
13	Rear Admiral / Major General
14	First Admiral / Brigadier General
15	Colonel
16	Lieutenant Colonel
17	Major
18	Captain
19	First Lieutenant
20	Second Lieutenant
21	Candidate Officer

03. Military Education Level MEDLEV

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Education level	MEDLEVID	C	2	Education level
Description	MEDLEDES	C	30	Description

11 National Defence
12 Joint Command & Staff College
13 Command and Staff College
14 Staff College
15 2nd Off. Continuing Education
16 1st Off. Continuing Edu.
17 Naval Academy
18 Basic Officer School
19 Candidate Officer School

04. General Education Level CEDLEV

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Edication level	CEDLEVEL	C	2	Education level
Description	CEDUDES	C	30	Level of Education

11 University Level
12 Academy Level
13 Senior High School Level
14 Junior High School Level
15 Elementary School Level

05. Table of Religion Code RELIGION

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Religion	RELCODE	N	1	Religion code
Description	RELDES	C	10	Name of Religion

1 Moslem
2 Catholic
3 Protestant
4 Budhist
5 Hindu
6 Other

06. Table of Sex Code SEXCODE

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Sex	SEX	N	1	Sex Code
Sex	SEXDES	C	6	Sex

1 Male	2 Female

07. Table of City code CITYCODE

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
City code	CITYCODE	N	3	City code
Description	CITYDES	C	20	The name of City

101	Jakarta	201	Tokyo
102	Bandung	201	Bangkok
103	Bogor	203	Singapore
104	Semarang	203	Manila
105	Jogyakarta	214	Paris
106	Surakarta	215	London
107	Magelang	301	San Francisco
108	Surabaya	302	San Diego
109	Malang	303	Monterey
110	Madiun	304	Washington DC
111	Medan	305	Quantico
112	Ujungpandang	306	Newport
113	Denpasar	307	Lackland
	etc.		etc.

08. Table of Education Result RESULT

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Result code	RESCODE	N	1	Education result code
Description	RESDES	C	25	Result of education

1 Graduate	3 Incomplete
2 Certificate of Completion	

09. Table of Skin Color COLORSK

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Skin code	SKINCODE	N	1	Skin color code
Description	SKINDES	C	6	Skin color description

1 Brown	3 White
2 Black	4 Yellow

10. Table of Hair Type TYPEHR

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Hair code	HAIRCODE	N	1	Hair code
Description	HAIRSDES	C	15	Hair description

1 Curly	3 Straight stiff
2 Wavy	4 Straight limp

11. Table of Eye Color COLOREYE

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Eye code	EYECODE	N	1	Eye color code
Description	EYEDES	C	5	Eye color

1 Black	3 Brown
2 Blue	

12. Table of Blood Types BLOOD

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Blood code	BLOODCOD	N	1	Blood code
Description	BLOODDES	C	2	Blood description

1	A+	5	AB+
2	A-	6	AB-
3	B+	7	O +
4	B-	8	O -

13. Table of Uniform Size UNIFORM

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Uniform code	UNICODE	N	1	Uniform code
Description	UNIDES	C	6	Uniform description

1	Small	3	Large
2	Medium		

14. Table of Echelon Code ECHELON

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Echelon code	ECHCODE	C	2	Echelon code
Description	ECHDES	C	11	Echelon description

11	Echelon 1-A	23	Echelon 2-C
12	Echelon 1-B	24	Echelon 2-D
13	Echelon 1-C	25	Echelon 2-E
14	Echelon 1-D	26	Echelon 2-F
15	Echelon 1-E	31	Echelon 3-A
16	Echelon 1-F	32	Echelon 3-B
17	Echelon 1-G	33	Echelon 3-C
18	Echelon 1-H	34	Echelon 3-D
21	Echelon 2-A	35	Echelon 3-E
22	Echelon 2-B	41	Functional

15. Occupation list JOBLIST

DATA ITEM	ACRONYM	TYPE	LENGTH	DESCRIPTION
Job ID	JOBID	C	3	Job code
Job Descript	JOBDES	C	30	Name of Job
Echelon	ECHELON	C	2	Job echelon (see ECHELON table 14)
Station	JOBSTAT	C	2	Station

APPENDIX D

EXAMPLES OF PROGRAM

```
***************************************************************
*  PROGRAM NAME : PERS.PRG                                    *
*  AUTHOR       : DJOKO M.  ARIYADI                           *
*  DATE WRITTEN : MAY, 1986                                   *
*  PURPOSE      : THIS IS THE MAIN MENU PROGRAM FOR THE       *
*                 OFFICER SYSTEM                              *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                      *
*                 ADDMENU.PRG          DELMENU.PRG            *
*                 CHGMENU.PRG          PRNMENU.PRG            *
*                 QUEMENU.PRG                                 *
*  NO FILES DBF ARE USED.                                     *
***************************************************************
SET COLOR TO 30,14
SET TALK OFF
SET CONSOLE ON
ERASE
STORE '    ' TO PASS
@ 8,22  SAY ' WELCOME TO THE NAVY OFFICER PERSONNEL SYSTEM. '
@ 12,22 SAY '     ENTER YOUR PASSWORD PLEASE   '
SET CONSOLE OFF
ACCEPT TO PASS
STORE !(PASS) TO PASS
SET CONSOLE ON
IF TRIM(PASS) <> 'PERS'
@ 15,10 SAY 'YOU TYPED THE WRONG PASSWORD, TRY AGAIN FROM THE;
BEGINNING'
@ 17,25 SAY 'QUIT TO DBASE II !!'
CANCEL
ENDIF
RELEASE PASS

  DO WHILE T
    STORE " " TO ANSWER
    SET TALK OFF
    ERASE
    ? "  "
    @ 2,1 SAY  "OFFICER                         MAIN MENU      ;
                   MAY 1986"
     SET COLOR TO 4
    ? "========================================================;
======================"
    ? "  "
    ? "   "
    ? "                            A ---> ADD RECORD"
    ? "  "
    ? "                            D ---> DELETE RECORD"
    ? "        "
    ? "                            C ---> CHANGE RECORD"
    ? " "
    ? "                            P ---> PRINT"
    ? " "
    ? "                            Q ---> QUERY INFORMATION "
    ? "  "
    ? "                            R ---> RETURN TO DATABASE"
    ? "  "
    ? "                            S ---> RETURN TO DOS"
    ? "  "
    ? "  "
    ?"======================-=================================;
========================"

   @ 23,16 SAY "SELECT AN ACTION BY LETTER.=======> " GET ANSWER;
 PICTURE"!"

    @ 23,16 SAY CHR (7)
    READ
    DO CASE
      CASE ANSWER = 'A'
        DO ADDMENU
      CASE ANSWER = 'D'
        DO DELMENU
      CASE ANSWER = 'C'
        DO CHGMENU
      CASE ANSWER = 'P'
        DO PRNMENU
      CASE ANSWER = 'Q'
        DO QUEMENU
      CASE ANSWER = 'R'
        CANCEL
      CASE ANSWER = 'S'
        DO QUIT
      OTHERWISE
        DO ERROR
    ENDCASE
  ENDDO
ERASE
RETURN
```

```
*****************************************************************
*  PROGRAM NAME : ADDMENU.PRG                                   *
*  AUTHOR       : DJOKO M.  ARIYADI                             *
*  DATE WRITTEN : MAY, 1986                                     *
*  PURPOSE      : TO PRESENT THE ADDITION MENU                  *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                        *
*        ADDOFF.PRG    ADDRANK.PRG   ADDCORPS.PRG   ADDRCODE.PRG *
*        ADDJOB.PRG    ADDJCODE.PRG  ADDEDU.PRG     ADDECODE.PRG *
*  LOCAL VARIABLE USED : ANSWER                                 *
*  NO FILES DBF ARE USED.                                       *
*****************************************************************
CLEAR
ERASE
  DO WHILE T
    STORE " " TO ANSWER
    SET TALK OFF
    ERASE
    ? "    "
    @ 2,1 SAY "ADDITION                         MENU            ;
           MAY 1986"
     set color to 4
    ? "==========================================================;
==================="
    ? " "
    ? "                         RECORD ADDITION OPTION"
    ? "  "
    ? "  "
    ? "  "
    ? "          O --> OFFICER ADDITION                K --> JOB;
 ADDITION"
    ? "  "
    ? "          C --> CORPS ADDITION                  L --> JOB;
 CODE ADDITION "
    ? " "
    ? "          E --> EDUCATION ADDITION              R --> RANK;
 ADDITION "
    ? " "
    ? "          F --> EDUCATION CODE  ADDITION        S --> RANK;
 CODE ADDITION"
    ? "  "
    ? "   "
    ? " "
    ? "                       TYPE RETURN KEY, RETURN TO MENU"
    ?"===========================================================;
========================"

   @ 23,16 SAY "SELECT AN ACTION BY LETTER.=======> " GET ANSWER;
 PICTURE "!"
   @ 23,16 SAY CHR (7)
    READ
    DO CASE
      CASE ANSWER = ' '
```

```
        RETURN
      CASE ANSWER = 'O'
        DO ADDOFF
      CASE ANSWER = 'C'
        DO ADDCORPS
      CASE ANSWER = 'E'
        DO ADDEDU
      CASE ANSWER = 'F'
        DO ADDCEDU
      CASE ANSWER = 'K'
        DO ADDJOB
      CASE ANSWER = 'L'
        DO ADDCJOB
      CASE ANSWER = 'R'
        DO ADDRANK
      CASE ANSWER = 'S'
        DO ADDCRANK
      OTHERWISE
        DO ERROR
    ENDCASE
  ENDDO
ERASE
RETURN
```

```
****************************************************************
*  PROGRAM NAME : DELMENU.PRG                                  *
*  AUTHOR       : DJOKO M. ARIYADI                             *
*  DATE WRITTEN : MAY, 1986                                    *
*  PURPOSE      : TO PRESENT THE DELETE MENU                   *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                       *
*          DELOFF.PRG  DELRANK.PRG DELCORPS.PRG  DELRCODE.PRG  *
*
*          DELJOB.PRG  DELEDU.PRG DELCJOB.PRG  DELECODE.PRG    *
*  LOCAL VARIABLE USED : ANSWER                                *
*  NO FILES DBF ARE USED.                                      *
****************************************************************
CLEAR
ERASE
  DO WHILE T
    STORE " " TO ANSWER
    SET TALK OFF
    ERASE
    ? " "
    @ 2,1 SAY "DELETION                                MENU           ;
                MAY 1986"
     SET COLOR TO 4
    ? "=========================================================;
====================="
    ? " "
    ? "                             DELETE OPTION "
    ? "  "
    ? "  "
    ? "  "
    ? "        O --> DELETE OFFICER                  K --> DELETE;
 JOB "
    ? "  "
    ? "        C --> DELETE CORPS                    L --> DELETE;
 JOB CODE"
    ? " "
    ? "        E --> DELETE EDUCATION                R --> DELETE;
 RANK "
    ? " "
    ? "        F --> DELETE EDUCATION CODE           S --> DELETE;
 RANK CODE"
    ? "  "
    ? "   "
    ? " "
    ? "                         TYPE RETURN KEY, RETURN TO MENU"
    ?"==========================================================;
=================="

   @ 23,16 SAY "SELECT AN ACTION BY LETTER.=======> " GET ANSWER;
 PICTURE "!"

    @ 23,16 SAY CHR (7)
    READ
    DO CASE
```

```
      CASE ANSWER = ' '
        RETURN
      CASE ANSWER = 'O'
        DO DELOFF
      CASE ANSWER = 'C'
        DO DELCORPS
      CASE ANSWER = 'E'
        DO DELEDU
      CASE ANSWER = 'F'
        DO DELCEDU
      CASE ANSWER = 'K'
        DO DELJOB
      CASE ANSWER = 'L'
        DO DELCJOB
      CASE ANSWER = 'R'
        DO DELRANK
      CASE ANSWER = 'S'
        DO DELCRANK
      OTHERWISE
        DO ERROR
    ENDCASE
  ENDDO
ERASE
RETURN
```

```
**************************************************************
*  PROGRAM NAME : CHGMENU.PRG                                *
*  AUTHOR       : DJOKO M.  ARIYADI                          *
*  DATE WRITTEN : MAY, 1986                                  *
*  PURPOSE      : TO PRESENT THE CHANGE MENU                 *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                     *
*    CHGOFF.PRG    CHGRANK.PRG   CHGCORPS.PRG   CHGRCODE.PRG *
*    CHGJOB.PRG    CHGJCODE.PRG  CHGECODE.PRG   CHGEDU.PRG   *
*  LOCAL VARIABLE USED : ANSWER                              *
*  NO FILES DBF ARE USED.                                    *
**************************************************************
CLEAR
ERASE
  DO WHILE T
    STORE " " TO ANSWER
    SET TALK OFF
    ERASE
    ? " "
    @ 2,1 SAY  "CHANGE                              MENU                ;
           MAY 1986"
    SET COLOR TO 4
    ? "==========================================================;
==================="
    ? " "
    ? "                          RECORD CHANGE OPTION"
    ? "  "
    ? "  "
    ? "  "
    ? "        O --> CHANGE OFFICER            K --> CHANGE;
 JOB"
    ? "  "
    ? "        C --> CHANGE CORPS              L --> CHANGE;
 JOB CODE"
    ? " "
    ? "        E --> CHANGE EDUCATION          R --> CHANGE;
 RANK "
    ? " "
    ? "        F --> CHANGE EDUCATION CODE     S --> CHANGE;
 RANK CODE "
    ? "  "
    ? "   "
    ? " "
    ? "                       TYPE RETURN KEY, RETURN TO MENU"
    ?"==========================================================;
======================="

    @ 23,16 SAY "SELECT AN ACTION BY LETTER.=======> " GET ANSWER;
PICTURE "!"

    @ 23,16 SAY CHR (7)
    READ
    DO CASE
```

```
      CASE ANSWER = ' '
        RETURN
      CASE ANSWER = 'O'
        DO CHGOFF
      CASE ANSWER = 'C'
        DO CHGCORPS
      CASE ANSWER = 'E'
        DO CHGEDU
      CASE ANSWER = 'F'
        DO CHGCEDU
      CASE ANSWER = 'K'
        DO CHGJOB
      CASE ANSWER = 'L'
        DO CHGCJOB
      CASE ANSWER = 'R'
        DO CHGRANK
      CASE ANSWER = 'S'
        DO CHGCRANK
      OTHERWISE
        DO ERROR
    ENDCASE
  ENDDO
ERASE
RETURN
```

```
***************************************************************
*  PROGRAM NAME :. QUEMENU.PRG                                *
*  AUTHOR       : DJOKO M. ARIYADI                            *
*  DATE WRITTEN : MAY, 1986                                   *
*  PURPOSE      : TO PROVIDE QUERY MENU SELECTION             *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                      *
*     QUERYA, QUERYB, QUERYC, QUERYD, QUERYE.PRG              *
*  LOCAL VARIABLE USED : ANSWER                               *
*  NO FILES DBF ARE USED.                                     *
***************************************************************
CLEAR
ERASE
  DO WHILE T
    STORE " " TO ANSWER
    SET TALK OFF
    ERASE
    ? "  "
    @ 2,1 SAY "QUERY                          MENU                ;
               MAY 1986"
     SET COLOR TO 4
    ? "=========================================================;
==================="
    ? "                               QUERY OPTIONS "
    ? "  "
    ? "  "

    ? "  "
    ? "                A ---> PROVIDE MIL. EDUCATION LEVEL;
 INFORMATION"
    ? "  "
    ? "                B ---> PROVIDE ANY MIL. EDUCATION;
 INFORMATION"
    ? "  "
    ? "                C ---> PROVIDE RANK HISTORY INFRMATION"
    ? "  "
    ? "                D ---> PROVIDE JOB HISTORY INFORMATION"
    ? "  "
    ? "                E ---> PROVIDE JOB ELIGIBLE INFORMATION "
    ? "  "
    ? "  "
    ? "                        TYPE RETURN KEY, RETURN TO MENU"
    ?"=========================================================;
======================="

   @ 23,16 SAY "SELECT AN ACTION BY LETTER.=======> " GET ANSWER;
 PICTURE "!"

    @ 23,16 SAY CHR (7)
    READ
    DO CASE
      CASE ANSWER = ' '
        RETURN
      CASE ANSWER = 'A'
        DO QUERYA
      CASE ANSWER = 'B'
        DO QUERYB
      CASE ANSWER = 'C'
        DO QUERYC
      CASE ANSWER = 'D'
        DO QUERYD
      CASE ANSWER = 'E'
        DO QUERYE
      OTHERWISE
        DO ERROR
    ENDCASE
  ENDDO
ERASE
RETURN
```

```
****************************************************************
*  PROGRAM NAME : PRNMENU.PRG                                  *
*  AUTHOR       : DJOKC M. ARIYADI                             *
*  DATE WRITTEN : MAY, 1986                                    *
*  PURPOSE      : TO PROVIDE PRINT MENU SELECTION              *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                       *
*              PROFMENU, PRRKMENU, PRJOMENU.PRG                *
*  LOCAL VARIABLE USED : ANSWER                                *
*  NO FILES DBF ARE USED.                                      *
****************************************************************
ERASE
  DO WHILE T
    STORE " " TO ANSWER
    SET TALK OFF
    ERASE
    ? "  "
    @ 2,1 SAY "PRINT                               MENU            ;
              MAY 1986"
     SET COLOR TO 4
    ? "=================================================-===========;
===================="
    ? "                           PRINT MENU OPTIONS "
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ? "                            P ---> OFFICER RECORD"
    ? "  "
    ? "  "
    ? "                            R ---> RANK CODE RECORD"
    ? "  "
    ? "  "
    ? "                            J ---> JOB CODE RECORD"
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ? "                      TYPE RETURN KEY, RETURN TO MENU"
    ?"==============================================================;
===================="

   @ 23,16 SAY "SELECT AN ACTION BY LETTER.=======> " GET ANSWER;
 PICTURE "!"

    @ 23,16 SAY CHR (7)
    READ
    DO CASE
      CASE ANSWER = ' '
        RETURN
      CASE ANSWER = 'P'
        DO PROFMENU
      CASE ANSWER = 'R'
        DO PRRKMENU
      CASE ANSWER = 'J'
        DO PRJOMENU
      OTHERWISE
        DO ERROR
    ENDCASE
  ENDDO
ERASE
RETURN
```

```
**************************************************************
*  PROGRAM NAME : PROFMENU.PRG                               *
*  AUTHOR       : DJOKO M. ARIYADI                           *
*  DATE WRITTEN : MAY , 1986                                 *
*  PURPOSE      : TO PROVIDE OFFICER PRINT MENU PROGRAM      *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                     *
*  PERSNAME.FRM, PERSERNO.FRM, PERSRANK.FRM, PERSRKSN.FRM    *
*  LOCAL VARIABLE USED : ANSWER,ANSW                         *
*  NO FILES DBF ARE USED.                                    *
**************************************************************
CLEAR
SET TALK OFF
SET COLOR TO 20,5
DO WHILE T
   ERASE
   ? ' '
   @ 2,1 SAY ' PRINT OFFICERS                     OPTION          ;
             MAY 1986'
   SET COLOR TO 4
   ? '===========================================================;
=================='
   ? ' '
   ? '                        PRINT OFFICERS LIST OPTION'
   ? ' '
   ? ' '
   ? '                       N --->   LIST BY NAME '
   ? '  '
   ? '                       R --->   LIST BY RANK'
   ? '  '
   ? '                       S --->   LIST BY SERIAL # '
   ? '  '
   ? '  '
   ? '  '
   ? '  '
   ? ' '
   ? '                   TYPE RETURN KEY, RETURN TO MENU'
   SET COLOR TO 4
   ? '===========================================================;
====================='
      STORE ' ' TO ANSWER
      @  23,16  SAY  "  SELECT AN ACTION BY LETTER  ====>  " GET;
ANSWER PICTURE "!"
      READ
      DO CASE
         CASE ANSWER = ' '
           RETURN
         CASE ANSWER = 'N'
           DO PROFNAME
         CASE ANSWER = 'R'
           DO PROFRANK
         CASE ANSWER = 'S'
           DO PROFSERN
         OTHERWISE
           DO ERROR
      ENDCASE
ENDDO
ERASE
RETURN
```

```
***************************************************************
*   PROGRAM NAME : PROFRANK.PRG                               *
*   AUTHOR       : DJOKO M. ARIYADI                           *
*   DATE WRITTEN : MAY , 1986                                 *
*   PURPOSE      : TO PRINT OFFICER'S LISTING BY RANK         *
*   PROGRAMS CALLED BY THIS PROGRAM ARE :                     *
*                   PERSRANK.FRM                              *
*   LOCAL VARIABLE USED : Q:PRS                               *
*   DBF FILE USED ARE : OFFICER,RANKLIST                      *
***************************************************************
ERASE
CLEAR
SET TALK OFF
SET COLOR TO 20,5
STORE T TO Q:FLAG1
DO WHILE Q:FLAG1
    USE OFFICER

    @ 12,20 SAY " BE PATIENT PLEASE, STILL PROCESSING"
    INDEX ON RANKID TO TF1

    SELECT PRIMARY
    USE OFFICER INDEX TF1
    SELECT SECONDARY
    USE RANKLIST
    JOIN TO OFFRANK FOR P.RANKID=S.RANKID FIELDS P.SERNO,P.NAME, ;
RANKDES

    USE OFFRANK
    ERASE
    STORE T TO Q:FLAG2
    DO WHILE Q:FLAG2
      ERASE
      @ 8,15 SAY 'DO YOU WANT THIS LIST ON THE SCREEN or :
 PRINTER?'
      ? "  "
      ACCEPT "                  ENTER 's' for SCREEN or 'h' for :
 HARDCOPY" to Q:PRS
      READ
      DO CASE
         CASE Q:PRS = 's'
           ERASE
           REPORT FORM PERSRANK
           STORE F TO Q:FLAG1
           STORE F TO Q:FLAG2
         CASE Q:PRS = 'h'
           ERASE
           REPORT FORM PERSRANK TO PRINT
           STORE F TO Q:FLAG1
           STORE F TO Q:FLAG2
         OTHERWISE
            STORE F TO Q:FLAG1
            DO ERROR
            ERASE
            LOOP
       ENDCASE
     ENDDO
     SET COLOR TO 4
     ? "  "
     ? "                     TYPE ANY KEY TO CONTINUE "
     SET CONSOLE OFF
     WAIT
     SET CONSOLE ON
ENDDO
RETURN
```

```
*************************************************************
*  PROGRAM NAME : PROFNAME.PRG                              *
*  AUTHOR       : DJOKO M. ARIYADI                          *
*  DATE WRITTEN : MAY , 1986                                *
*  PURPOSE      : TO PRINT OFFICER'S LISTING BY NAME        *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                    *
*                  PERSNAME.FRM                             *
*  LOCAL VARIABLE USED : Q:PRS                              *
*  DBF FILE USED ARE : OFFICER,RANKLIST                     *
*************************************************************
ERASE
CLEAR
SET TALK OFF
SET COLOR TO 20,5
STORE T TO Q:FLAG1
DO WHILE Q:FLAG1
    USE OFFICER

    @ 12,20 SAY " BE PATIENT PLEASE, STILL PROCESSING"
    INDEX ON NAME TO TF1

    SELECT PRIMARY
    USE OFFICER INDEX TF1
    SELECT SECONDARY
    USE RANKLIST
    JOIN TO OFFRANK FOR P.RANKID=S.RANKID FIELDS P.SERNO,P.NAME,;
RANKDES

    USE OFFRANK
    ERASE
    STORE T TO Q:FLAG2
    DO WHILE Q:FLAG2
      ERASE
      @ 8,15 SAY 'DO YOU WANT THIS LIST ON THE SCREEN or PRINTER?'
      ? "  "
      ACCEPT  "                  ENTER 's' tor SCREEN or 'h' for;
 HARDCOPY" to Q:PRS
      READ
      DO CASE
         CASE Q:PRS = 's'
           ERASE
           REPORT FORM PERSNAME
           STORE F TO Q:FLAG1
           STORE F TO Q:FLAG2
         CASE Q:PRS = 'h'
           ERASE
           REPORT FORM PERSNAME TO PRINT
           STORE F TO Q:FLAG1
           STORE F TO Q:FLAG2
         OTHERWISE
            STORE F TO Q:FLAG1
            DO ERROR
            ERASE
            LOOP
       ENDCASE
     ENDDO
     SET COLOR TO 4
     ? "  "
     ? "                     TYPE ANY KEY TO CONTINUE "
     SET CONSOLE OFF
     WAIT
     SET CONSOLE ON
ENDDO
RETURN
```

```
*****************************************************************
*  PROGRAM NAME : PROFSERN.PRG                                  *
*  AUTHOR       : DJOKO M. ARIYADI                              *
*  DATE WRITTEN : MAY, 1986                                     *
*  PURPOSE      : TO PRINT OFFICER'S LISTING BY SERIAL NUMBER   *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                        *
*                  PERSERNO.FRM                                 *
*  LOCAL VARIABLE USED : Q:PRS                                  *
*  DBF FILE USED ARE : OFFICER,RANKLIST                         *
*****************************************************************
ERASE
CLEAR
SET TALK OFF
SET COLOR TO 20,5
STORE T TO Q:FLAG1
DO WHILE Q:FLAG1
    USE OFFICER

    @ 12,20 SAY " BE PATIENT PLEASE, STILL PROCESSING"

    INDEX ON SERNO TO TF1

    SELECT PRIMARY
    USE OFFICER INDEX TF1
    SELECT SECONDARY
    USE RANKLIST
    JOIN TO OFFRANK FOR P.RANKID=S.RANKID FIELDS P.SERNO,P.NAME,;
RANKDES

    USE OFFRANK
    ERASE
    STORE T TO Q:FLAG2
    DO WHILE Q:FLAG2
      ERASE
      @ 8,15 SAY 'DO YOU WANT THIS LIST ON THE SCREEN or PRINTER?'
      ? "  "
      ACCEPT  "                    ENTER 's' for SCREEN or 'h' for;
 HARDCOPY" to Q:PRS
      READ
      DO CASE
         CASE Q:PRS = 's'
           ERASE
           REPORT FORM PERSERNO
           STORE F TO Q:FLAG1
           STORE F TO Q:FLAG2
         CASE Q:PRS = 'h'
           ERASE
           REPORT FORM PERSERNO TO PRINT
           STORE F TO Q:FLAG1
           STORE F TO Q:FLAG2
         OTHERWISE
            STORE F TO Q:FLAG1
            DO ERROR
            ERASE
            LOOP
       ENDCASE
     ENDDO
     SET COLOR TO 4
     ? "  "
     ? "                  TYPE ANY KEY TO CONTINUE "
     SET CONSOLE OFF
     WAIT
     SET CONSOLE ON
ENDDO
RETURN
```

```
*************************************************************
*  PROGRAM NAME : DELOFF.PRG                                *
*  AUTHOR       : DJOKO M. ARIYADI                          *
*  DATE WRITTEN : MAY,1986                                  *
*  PURPOSE      : TO DELETE OFFICER RECORD                  *
*  PROGRAMS CALLED BY THIS PROGRAM ARE                      *
*  DBF FILES USED ARE : OFFICER                             *
*************************************************************
SET TALK OFF
SET CONSOLE ON
ERASE
STORE t TO flag
DO WHILE t
 SET COLOR TO 30,14

 USE OFFICER
 INDEX ON SERNO TO DELOFF

 USE OFFICER INDEX DELOFF
      STORE '      ' TO msn
      ERASE
      @ 5,23 SAY  'TO DELETE OFFICER RECORD'
      @ 10,23 SAY 'ENTER SERIAL NUMBER ' GET msn PICTURE '999999'
      @ 12,23 SAY 'e.g : 005001; 006005; 007001'
      @ 16,23 SAY 'ENTER A BLANK, RETURN TO DELETE MENU '
      READ
       if msn = '      '
        RELEASE msn,flag
        RETURN
       endif
      FIND &msn
      IF # = 0
            SET COLOR TO 112,140
            @ 12,20 SAY 'SERIAL NUMBER NOT ON THIS PERS RECORDS'
            @ 13,40 SAY CHR(7)
            SET TALK OFF
            STORE 1 TO v
            DO WHILE v<200
                STORE v+1 TO v
            ENDDO WHILE v<250
            SET COLOR TO 30,14
            LOOP
      ENDIF # = 0
      ERASE

      STORE '                    ' to mname
      STORE '  ' to mcorp
      STORE  0 to mrankid
      STORE '  ' to meduid
      STORE ' ' to msex
      STORE '  ' to mbirthdd
      STORE '  ' to mbirthmm
      STORE '    ' to mbirthyy
      STORE '                    ' to mbirthp
      STORE ' ' to mreligion

      STORE serno to msn
      STORE name to mname
      STORE corps to mcorps
      STORE rankid to mrankid
      STORE eduid to meduid
      STORE sex to msex
      STORE birthdd to mbirthdd
      STORE birthmm to mbirthmm
      STORE birthyy to mbirthyy
      STORE birthp to mbirthp
      STORE religion to mreligion

      @ 4,20 say ' THIS IS THE DATA YOU WANT TO DELETE'
      @ 7,23 SAY 'SERIAL #             '+msn
      @ 8,23 SAY 'NAME                 'GET mname
      @ 9,23 SAY 'CORPS                'GET mcorps
      @ 10,23 SAY 'RANK                 'GET mrankid
      @ 11,23 SAY 'MIL. EDUCATION       'GET meduid
      @ 12,23 SAY 'SEX                  'GET msex
      @ 13,23 SAY 'BIRTH DAY            'GET mbirthdd
      @ 14,23 SAY 'BIRTH MONTH          'GET mbirthmm
      @ 15,23 SAY 'BIRTH YEAR           'GET mbirthyy
      @ 16,23 SAY 'BIRTH PLACE          'GET mbirthp
      @ 17,23 SAY 'RELIGION             'GET mreligion

      STORE ' ' TO answer
      @ 19,20 SAY ' ARE YOU SURE TO DELETE THIS DATA ?'
      @ 21,16 SAY ' TYPE y FOR yes OR n RETURN TO SERIAL #;
 OPTION'
      @ 21,65 GET answer
      READ
      DO CASE
         CASE (answer) = 'y'
            DELETE
            PACK
            CLEAR
            STORE t TO flag
         CASE (answer) = 'n'
            STORE t TO flag
         OTHERWISE
            DO ERROR
      ENDCASE
ENDDO WHILE t
USE OFFICER
INDEX ON SERNO TO DELOFF
ERASE
RETURN
```

```
*************************************************************
*  PROGRAM NAME : CHGOFF.PRG                                *
*  AUTHOR       : DJOKO M. ARIYAD                           *
*  DATE WRITTEN : MAY, 1986                                 *
*  PURPOSE      : TO CHANGE OFFICER  RECORDS                *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                    *
*  DBF FILES USED ARE : OFFICER                             *
*************************************************************
SET TALK OFF
SET CONSOLE ON
ERASE

USE OFFICER
INDEX ON SERNO TO CHGOFF
USE OFFICER INDEX CHGOFF

STORE t TO flag
DO WHILE t
  SET COLOR TO 30,14
      STORE '      ' TO msn
      ERASE
      @ 5,23 SAY  'TO CHANGE OFFICER RECORDS'
      @ 10,23 SAY 'ENTER SERIAL NUMBER ' GET msn PICTURE '999999'
      @ 12,23 SAY 'e.g : 005001; 006005; 007001'
      @ 16,23 SAY 'ENTER A BLANK RETURN TO CHANGE MENU '
      READ
       IF msn = '      '
        RELEASE msn,flag
        RETURN
       ENDIF
      FIND &msn
      IF # = 0
            SET COLOR TO 112,140
            @ 12,20 SAY 'SERIAL NUMBER NOT ON THIS OFFICER;
 RECORDS'
            @ 13,40 SAY CHR(7)
            SET TALK OFF
            STORE 1 TO v
            DO WHILE v<200
                STORE v+1 TO v
            ENDDO WHILE v<235
            SET COLOR TO 30,14
            LOOP
      ENDIF # = 0
      ERASE
      * load old values
      STORE name TO mname
      STORE corps TO mcorps
      STORE rankid TO mrankid
      STORE eduid TO meduid
      STORE sex TO msex
      STORE birthdd TO mbirthdd
      STORE birthmm TO mbirthmm
      STORE birthyy TO mbirthyy
      STORE birthp TO mbirthp
      STORE religion TO mreligion
      * get new values
      @ 4,20 say ' THIS IS THE DATA YOU WANT TO CHANGE'
      @ 7,23 SAY 'SERIAL NUMBER   '+msn
      @ 8,23 SAY 'NAME            'GET mname
      @ 9,23 SAY 'CORPS           'GET mcorps
      @ 10,23 SAY 'RANK            ' GET mrankid
      @ 11,23 SAY 'MIL. EDUCATION  ' GET meduid
      @ 12,23 SAY 'SEX             'GET msex
      @ 13,23 SAY 'BIRTH DATE      'GET mbirthdd
      @ 14,23 SAY 'BIRTH MONTH     'GET mbirthmm
      @ 15,23 SAY 'BIRTH YEAR      'GET mbirthyy
      @ 16,23 SAY 'BIRTH PLACE     'GET mbirthp
      @ 17,23 SAY 'RELIGION        'GET mreligion

      STORE 'y' TO answer
      @ 20,20 SAY ' ARE YOU SURE TO CHANGE THIS DATA ?'
      @ 22,16 SAY 'TYPE y FOR yes OR n RETURN TO SERIAL # OPTION'
      @ 22,65 GET answer
      READ
      DO CASE
         * replace values
         CASE (answer) = 'y'
            REPLACE name WITH mname, corps WITH mcorps, rankid;
 WITH mrankid
            REPLACE eduid WITH meduid, sex WITH msex, birthdd;
 WITH mbirthdd
            REPLACE birthmm WITH mbirthmm, birthyy WITH mbirthyy
            REPLACE birthp WITH mbirthp, religion WITH mreligion
            STORE t to flag
         CASE (answer) = 'n'
            STORE t TO flag
         OTHERWISE
            DO ERROR
      ENDCASE
ENDDO WHILE t
USE OFFICER
INDEX ON SERNO TO CHGOFF
ERASE
RETURN
```

```
************************************************************
*  PROGRAM NAME : ADDOFF.PRG                               *
*  AUTHOR       : DJOKO M. ARIYADI                         *
*  DATE WRITTEN : MAY, 1986                                *
*  PURPOSE      : TO PERMIT ADDITIONS TO OFFICER RECORDS   *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                   *
*  DBF FILES USED ARE : OFFICER                            *
************************************************************
SET TALK OFF
SET CONSOLE ON
ERASE
USE OFFICER
INDEX ON SERNO TO ADDOFF
USE OFFICER INDEX ADDOFF
STORE t TO flag
DO WHILE T
 SET COLOR TO 30,14

      STORE '      ' TO msn
      ERASE
      @ 5,20 SAY  'ADDITION TO OFFICER RECORDS'
      @ 10,18 SAY 'ENTER SERIAL NUMBER ' GET msn PICTURE;
 '999999'
      @ 12,18 SAY 'e.g : 005004; 007002; 009001'
      @ 16,16 SAY 'ENTER A BLANK TO RETURN TO ADDITION MENU'
      READ
       IF msn = '    '
        RELEASE msn,flag
        RETURN
       ENDIF
      FIND &msn
      IF #<>0
            SET COLOR TO 112,140
            @  12,10  SAY  'THIS  SERIAL # IS  ALREADY ON FILE'

            @ 13,40 SAY CHR(7)
            SET TALK OFF
            STORE 1 TO v
            DO WHILE v<200
                STORE v+1 TO v
            ENDDO WHILE v<250
            SET COLOR TO 30,14
            LOOP
      ENDIF #<>0
      ERASE
      @ 4,23 SAY 'ADDING TO OFFICER RECORD'
      STORE '                    ' TO mname
      STORE '  ' TO mcorps
      STORE 0 TO mrankid
      STORE '  ' TO meduid
      STORE ' ' TO msex
      STORE '  ' TO mbirthdd
      STORE '  ' TO mbirthmm
      STORE '    ' TO mbirthyy
      STORE '                    ' TO mbirthp
      STORE ' ' TO mreligion
      @ 7,23 SAY 'SERIAL NUMBER      ' +msn
      @ 8,23 SAY 'NAME               ' GET mname
      @ 9,23 SAY 'CORPS              ' GET mcorps
      @ 10,23 SAY 'RANK               ' GET mrankid
      @ 11,23 SAY 'MIL. EDUCATION     ' GET meduid
      @ 12,23 SAY 'SEX                ' GET msex
      @ 13,23 SAY 'BIRTH DAY          ' GET mbirthdd
      @ 14,23 SAY 'BIRTH MONTH        ' GET mbirthmm
      @ 15,23 SAY 'BIRTH YEAR         ' GET mbirthyy
      @ 16,23 SAY 'BIRTH PLACE        ' GET mbirthp
      @ 17,23 SAY 'RELIGION           ' GET mreligion

      @ 20,10 say ' HIT ENTER TO RETURN TO OFFICER MENU OPTION;

      READ

      IF mname = '                         '
         LOOP
      ENDIF
      * add the record
      APPEND BLANK
      REPLACE serno WITH msn ,name WITH mname ,corps WITH mcorps
      REPLACE rankid WITH mrankid,eduid WITH meduid,sex;
 WITH msex
      REPLACE birthdd WITH mbirthdd ,birthmm WITH mbirthmm
      REPLACE birthyy WITH mbirthyy ,birthp WITH mbirthp
      REPLACE religion WITH mreligion

ENDDO WHILE t
USE OFFICER
INDEX ON SERNO TO ADDOFF
ERASE
RETURN
```

```
**************************************************************
*  PROGRAM NAME : QUERYA.PRG                                 *
*  AUTHOR       : DJOKO M. ARIYADI                           *
*  DATE WRITTEN : MAY, 1986                                  *
*  PURPOSE      : TO LIST OFFICERS WHO HAVE ACHIEVED A GIVEN *
*                 MILITARY EDUCATION LEVEL                   *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                     *
*  DBF FILES USED ARE : MILEDU. MEDULIST,OFFICER, CORPS,RANK *
**************************************************************
SET TALK OFF
ERASE
SET CONSOLE ON
STORE T TO FLAG
SET COLOR TO 30,14
STORE T TO Q:FLAG
DO WHILE Q:FLAG
    ERASE
    ? "  "
    @ 2,1 SAY "QUERY                  MILITARY EDUCATION LEVEL;
                     MAY 1986"
    SET COLOR TO 4
    ? "========================================================;
========================"
    SET COLOR TO 30,14
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ?  "    Given a Military Education Level, this program;
  will respond with :"
    ? "  "
    ? "               NAME, RANK, EDUCATION LEVEL, YEAR-END"
    ? "  "
    ? "  "
    ? "  The following is a list of Military Education Levels;
and their codes "
    ? "  "
    ? "             You can query one at a time "
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    SET COLOR TO 4
    ?"=========================================================;
======================="
    ? "  "
    ? "                     TYPE ANY KEY TO CONTINUE"
    SET CONSOLE OFF
    WAIT
    SET CONSOLE ON
    ERASE
    DO WHILE Q:FLAG
      ERASE
    @ 2,1 SAY "QUERY                   MILITARY EDUCATION LEVEL;
                        MAY 1986"
      SET COLOR TO 4
      ? "=======================================================;
=========================="
      SET COLOR TO 30,14
      USE MEDLEV
      ? "         MIL. EDUCATION #           DESCRIPTION"
      ? "  "
      LIST "                  " + MEDLEVID + "             " +;
 MEDLEDES  OFF
      SET COLOR,TO 30,14
      ? "  "
      ? "  "
      TEXT
         PLEASE SELECT ONE DESIRED MIL. EDUCATION NUMBER,;
 ACCORDING TO
        THE ABOVE LIST (2 DIGITS PLEASE) e.g. "15"  AND HIT;
 "RETURN" KEY
      ENDTEXT
      SET COLOR TO 4
      ? "  "
      ? "           TO FINISH (RETURN TO QUERY MENU OPTION) HIT;
 'RETURN'"
      ? "  "
      ? "  "
      SET COLOR TO 30,14
      ACCEPT "                    ENTER MIL. EDUCATION NUMBER;
 ( 2 DIGITS ) " TO Q:MNO

      ERASE
      @ 12,20 SAY "BE PATIENT PLEASE, STILL PROCESSING"
      IF Q:MNO = " "
        RELEASE ALL LIKE Q*
        ERASE
        STORE F TO Q:FOUND
        STORE F TO Q:FLAG
      ELSE
        STORE F TO Q:FOUND
        USE MEDLEV
        INDEX ON MEDLEVID TO QUERY1
        SET INDEX TO QUERY1
        FIND "&Q:MNO"
      IF #=0
         ERASE
         SET COLOR TO 112,140
         @ 10,9 SAY "               SORRY SCHOOL CANNOT BE FOUND"
         ? " "
         ? "          PLEASE, SEE THE MIL. EDUCATION NUMBER;
 LIST, AND TRY AGAIN"
```

```
         ? " "
         ? "                            TYPE ANY KEY, TC CONTINUE"
         SET CONSOLE OFF
         WAIT
         SET CONSOLE ON
         ERASE
         STORE F TO Q:FOUND
      ELSE
         STORE T TO Q:FOUND
      ENDIF
      ENDIF
      IF Q:FOUND
         STORE MEDLEDES TO Q:MEDLEV
         ERASE
         STORE MEDLEVID TO Q:MNO
         USE MEDLEV
         COPY TO TF1 FOR MEDLEVID = "&Q:MNO"

         USE TF1
         SELECT SECONDARY
         USE MILEDU
  JOIN TO TF2 FOR P.MEDLEVID=S.MEDLEVID FIELDS S.MEDUID,SERNO,;
MYYEDED,MEDURANK

         USE TF2
         SELECT SECONDARY
         USE MEDULIST
  JOIN TO TF3 FOR P.MEDUID=S.MEDUID FIELDS MEDUDES,P.SERNO,;
P.MYYEDED,P.MEDURANK

         USE TF3
         SELECT SECONDARY
         USE OFFICER
  JOIN TO TF4 FOR P.SERNO=S.SERNO FIELDS SERNO,NAME,RANKID,;
P.MEDURANK,P.MEDUDES,P.MYYEDED,CORPS

         USE TF4
         SELECT SECONDARY
         USE CORPCODE
  JOIN TO TF5 FOR P.CORPS=S.CORPS FIELDS CORPSDES,P.MEDURANK,;
P.SERNO,P.NAME,P.RANKID,P.MEDUDES,P.MYYEDED

         USE TF5
         SELECT SECONDARY
         USE RANKLIST
  JOIN TO TF6 FOR P.RANKID=S.RANKID FIELDS RANKDES,P.SERNO,;
P.NAME,P.MEDUDES,P.MYYEDED,P.CORPSDES

         STORE T TO Q:FLAG2
         DO WHILE Q:FLAG2

         @ 10,8 SAY "     DO YOU WANT TC SEE THE LIST ON ;
```

```
SCREEN or HARDCOPY ?"
         ?"  "
         ACCEPT "                 ENTER 's' FOR SCREEN or 'h';
 FOR HARDCOPY" TO Q:PRN
         READ
           DO CASE
              CASE Q:PRN = 's'
                 STORE F TO Q:FLAG2
                 ERASE
              CASE Q:PRN = 'h'
                 STORE F TO Q:FLAG2
                 ERASE
                 SET PRINT ON
                 ERASE
              OTHERWISE
                 DO ERROR
                 ERASE
                 STORE T TO Q:FLAG2
                 LOOP
           ENDCASE
        ENDDO

         USE TF1
         ? "                          EQUIVALENT WITH THE LEVEL;
 OF "
         LIST "                           " + MEDLEDES OFF
         ?  "   "
         ? " NAME                 RANK                        ;
 SCHOOL                YEAR-END"
         ? " =================================================;
================================"
         use tf6
         LIST " " + NAME + " " + RANKDES + " " + MEDUDES + " " ;
MYYEDED OFF
          SET COLOR TO 4
         ? " =================================================;
==============================="
         ? "                           TYPE ANY KEY TO CONTINUE"
         SET CONSOLE OFF
         WAIT
         SET CONSOLE ON
         SET PRINT OFF
      ENDIF
    ENDDO
    ERASE
ENDDO
ERASE
RETURN
```

```
**************************************************************
*  PROGRAM NAME : QUERYB.PRG                                 *
*  AUTHOR       : DJOKO M. ARIYADI                           *
*  DATE WRITTEN : MAY, 1986                                  *
*  PURPOSE      : TO LIST OFFICERS CURRENTLY ATTENDING       *
*                 MILITARY SCHOOL                            *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                     *
*  DBF FILES USED ARE : MILEDU. MEDULIST,OFFICER, CORPS,RANK *
**************************************************************
SET TALK OFF
ERASE
SET CONSOLE ON
STORE T TO FLAG
SET COLOR TO 30,14
STORE T TO Q:FLAG
DO WHILE Q:FLAG
    ERASE
    ? "  "
    @ 2,1 SAY "QUERY                   TYPES OF MILITARY;
 SCHOOL                     MAY 1986"
    SET COLOR TO 4
    ? "=======================================================;
========================"
    SET COLOR TO 30,14
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ? "    Given a Specific Military School. This program will;
 respond with : "
    ? "  "
    ? "              SERNO, NAME, RANK, NAME OF SCHOOL,;
 YEAR-START"
    ? "  "
    ? "  "
    ? "        The following is a list of Military School and;
 their codes"
    ? "  "
    ? "               You can query one code at a time "
 know"
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    SET COLOR TO 4
    ?"========================================================;
======================="
    ? "  "
    ? "                        TYPE ANY KEY TO CONTINUE"
    SET CONSOLE OFF
    WAIT
    SET CONSOLE ON
    ERASE
    DO WHILE Q:FLAG
      ERASE
    @ 2,1 SAY "QUERY                    TYPES OF MILITARY;
 SCHOOLS                    MAY 1986"
      SET COLOR TO 4
      ? "======================================================;
==========================="
      SET COLOR TO 30,14
      USE MEDULIST
      ? "          MIL. SCHOOL #              DESCRIPTION"
      ? "  "
      LIST "                   " + MEDUID + "             " + ;
MEDUDES  OFF
      SET COLOR TO 30,14
      ? "  "
      ? "  "
      TEXT
         PLEASE SELECT ONE DESIRED MIL. SCHOOL NUMBER,;
 ACCORDING TO
       THE ABOVE LIST (2 DIGITS PLEASE) e.g. "06"  AND HIT;
 "RETURN" KEY
      ENDTEXT
      SET COLOR TO 4
      ? "  "
      ? "               TO FINISH (RETURN TO QUERY MENU OPTION);
 HIT 'RETURN'"
      ? "  "
      ? "  "
      SET COLOR TO 30,14
      ACCEPT "                   ENTER MIL. SCHOOL NUMBER;
 ( 2 DIGITS ) " TO Q:MNO

      ERASE
      @ 12,20 SAY "BE PATIENT PLEASE, STILL PROCESSING"
      IF Q:MNO = " "
        RELEASE ALL LIKE Q*
        ERASE
        STORE F TO Q:FOUND
        STORE F TO Q:FLAG
      ELSE
        STORE F TO Q:FOUND
        USE MEDULIST
        INDEX ON MEDUID TO QUERY2
        SET INDEX TO QUERY2
        FIND "&Q:MNO"
      IF #=0
         ERASE
         SET COLOR TO 112,140
         @ 10,9 SAY "                 SORRY, SCHOOL CANNOT BE;
 FOUND"
         ? "  "
         ? "          PLEASE SEE THE MIL. SCHOOL NUMBER LIST;
 AND TRY ANOTHER NUMBER"
         ? "  "
         ?"                         TYPE ANY KEY, TO;
 CONTINUE"
         SET CONSOLE OFF
         WAIT
         SET CONSOLE ON
         ERASE
         STORE F TO Q:FOUND
      ELSE
         STORE T TO Q:FOUND
      ENDIF
      ENDIF
      IF Q:FOUND
         STORE MEDUDES TO Q:MEDLST
         ERASE
         STORE MEDUID TO Q:MNO
         USE MEDULIST
         COPY TO TF1 FOR MEDUID = "&Q:MNO"

         USE TF1
         SELECT SECONDARY
         USE MILEDU
  JOIN TO TF2 FOR P.MEDUID=S.MEDUID FIELDS SERNO,MYYEDST,;
MEDURANK

         USE TF2
         SELECT SECONDARY
         USE OFFICER
  JOIN TO TF3 FOR P.SERNO=S.SERNO FIELDS SERNO,NAME,CORPS,;
RANKID,P.MEDURANK,P.MYYEDST

         USE TF3
         SELECT SECONDARY
         USE CORPCODE
  JOIN TO TF4 FOR P.CORPS=S.CORPS FIELDS CORPSDES,P.SERNO,;
P.NAME,P.RANKID,P.MEDURANK,P.MYYEDST

         USE TF4
         SELECT SECONDARY
         USE RANKLIST
  JOIN TO TF5 FOR P.RANKID=S.RANKID FIELDS RANKDES,P.CORPSDES,;
P.SERNO,P.NAME,P.MYYEDST

         STORE T TO Q:FLAG2
         DO WHILE Q:FLAG2

         @ 10,8 SAY "     DO YOU WANT TO SEE THE LIST ON SCREEN;
 or HARDCOPY ?"
         ?"  "
         ACCEPT "                ENTER 's' FOR SCREEN or 'h';
 FOR HARDCOPY" TO Q:PRN
         READ
           DO CASE
              CASE Q:PRN = 's'
                 STORE F TO Q:FLAG2
                 ERASE
              CASE Q:PRN = 'h'
                 STORE F TO Q:FLAG2
                 ERASE
                 SET PRINT ON
                 ERASE
              OTHERWISE
                 DO ERROR
                 ERASE
                 STORE T TO Q:FLAG2
                 LOOP
           ENDCASE
        ENDDO

         USE TF1

         LIST "                                " + MEDUDES CFF
         ? "                         THE NAMES OF OFFICERS ARE "

         ? "  "
         ? " SERNO  RANK                        NAME                     ;
CORPS             YEAR-START"
         ? " ===================================================;
=============================="
         use tf5
         LIST " " + SERNO + " " + RANKDES + " " + NAME + " " +;
 CORPSDES + "      " MYYEDST OFF
          SET COLCR TO 4
         ? " ===================================================;
=============================="
         ? "                             TYPE ANY KEY TO CONTINUE"
         SET CONSOLE OFF
         WAIT
         SET CONSOLE ON
         SET PRINT OFF
      ENDIF
    ENDDO
    ERASE
ENDDO
ERASE
RETURN
```

```
****************************************************************
*  PROGRAM NAME : QUERYC.PRG                                   *
*  AUTHOR       : DJOKO M. ARIYADI                             *
*  DATE WRITTEN : MAY, 1986                                    *
*  PURPOSE      : GIVEN A SERIAL NUMBER, TO DESCRIBE THAT      *
*                 THAT OFFICER'S PROMOTION HISTORY             *
*  PROGRAMS CALLED BY THIS PROGRAM ARE :                       *
*  DBF FILES USED ARE : OFFICER, CORPS, RANK, RANKLIST         *
****************************************************************
SET TALK OFF
ERASE
SET CONSOLE ON
STORE T TO FLAG
SET COLOR TO 30,14
STORE T TO Q:FLAG
DO WHILE Q:FLAG
    ERASE
    ? "   "

    @ 2,1 SAY "QUERY                OFFICER'S PROMOTION HISTORY;
                    MAY 1986"

    SET COLOR TO 4
    ? "========================================================;
========================="
    SET COLOR TO 30,14
    ? "   "
    ? "   "
    ? "   "
    ? "   "
    ? "    Given an Officer's Serial Number, this program;
  will respond with :"
    ? "   "
    ? "      HIS PROMOTION HISTORY, INCLUDING DATE AND;
 ORDER NUMBER "
    ? "   "
    ? "   "
    ? "       The following is a list of Officer's Serial;
 Number and it's name "
    ? "   "
    ? "              You can type serial number which you like;
 to know"
    ? "   "
    ? "   "
    ? "   "
    ? "   "
    SET COLOR TO 4
    ?"=========================================================;
======================="
    ? "   "
    ? "                      TYPE ANY KEY TO CONTINUE"
    SET CONSOLE OFF
    WAIT



    SET CONSOLE ON
    ERASE
    DC WHILE Q:FLAG
      ERASE
    @ 2,1 SAY "QUERY                      OFFICER'S RANK ;
HISTORY                    MAY 1986"
      SET COLOR TO 4
      ? "=====================================================;
============================"
      SET COLOR TO 30,14
      USE OFFICER
      INDEX CN SERNO TO DXOFF
      SET INDEX TO DXOFF
      ? "                OFFICER'S SERIAL #, ALREADY ON THIS;
 FILE"
      ? "  "
      ? "                         SERIAL #      NAME"
      ? "  "
      LIST "                         " + SERNO + "     " + NAME;
 OFF
      SET COLOR TO 30,14
      ? "  "
      ? "  "
      TEXT
         PLEASE SELECT ONE DESIRED MIL. EDUCATION NUMBER, ;
ACCORDING TO
        THE ABOVE'LIST (2 DIGITS PLEASE) e.g. "06"  AND HIT;
 "RETURN" KEY
      ENDTEXT
      SET COLOR TO 4
      ? "  "
      ? "             TO FINISH (RETURN TO QUERY MENU OPTION);
HIT 'RETURN'"
      ? "  "
      SET COLOR TC 30,14
      ACCEPT "                   ENTER SERIAL NUMBER e.q:
 ( 6 DIGITS ) " TO Q:MNO
      ERASE
      @ 12,20 SAY "BE PATIENT PLEASE, STILL PROCESSING"
      IF Q:MNC = " "
        RELEASE ALL LIKE Q*
        ERASE
        STORE F TO Q:FOUND
        STORE F TO Q:FLAG
      ELSE
        STORE F TO Q:FOUND
        FIND "&Q:MNO"
      IF #=0
         ERASE
         SET COLOR TO 112,140
         @ 10,9 SAY "             SORRY, SERIAL NUMBER CANNOT BE;
 FOUND"
         ? "   "
         ?"              PLEASE, SEE THE SERIAL NUMBER'S LIST;
 AND TRY AGAIN"
         ? "   "
         ?"                         TYPE ANY KEY, TO CONTINUE"
         SET CONSOLE OFF
         WAIT
         SET CONSOLE ON
         ERASE
         STORE F TO Q:FOUND
      ELSE
         STORE T TO Q:FOUND
      ENDIF
      ENDIF
      IF Q:FOUND
         STORE NAME TO Q:OFF
         ERASE
         STORE SERNO TO Q:MNO
         USE OFFICER
         COPY TO TF1 FOR SERNO = "&Q:MNO"

         USE TF1
         SELECT SECONDARY
         USE RANKLIST
   JOIN TO TEMP1 FOR P.RANKID=S.RANKID FIELDS P.SERNO,P.NAME,;
RANKDES,P.CORPS

         USE TEMP1
         SELECT SECONDARY
         USE CORPCODE
   JOIN TO TEMP2 FOR P.CORPS=S.CORPS FIELDS P.SERNO,P.NAME,;
P.RANKDES,CORPSDES,P.NAME

         USE TF1
         SELECT SECONDARY
         USE RANK
  JOIN TO TF FOR P.SERNO=S.SERNO FIELDS S.RANKID,RANKDD,RANKMM,;
RANKYY,RORNO,RDATORNO

         USE TF
         SELECT SECONDARY
         USE RANKLIST
  JOIN TO TF3 FOR P.RANKID=S.RANKID FIELDS RANKDES,P.RANKDD,;
P.RANKMM,P.RANKYY,P.RORNO,P.RDATORNO
         USE TF3
         SORT ON RANKYY TO TF4 ASCENDING

         STORE T TO Q:FLAG2
         DO WHILE Q:FLAG2
         @ 10,8 SAY "     DO YOU WANT TO SEE THE LIST ON SCREEN ;
or HARDCOPY ?"
```

```
         ?"   "
         ACCEPT "                 ENTER 's' FOR SCREEN or 'h';
 FOR HARDCOPY" TO Q:PRN
         READ
           DO CASE
              CASE Q:PRN = 's'
                 STORE F TO Q:FLAG2
                 ERASE
              CASE Q:PRN = 'h'
                 STORE F TO Q:FLAG2
                 ERASE
                 SET PRINT ON
                 ERASE
              OTHERWISE
                 DO ERROR
                 ERASE
                 STORE T TO Q:FLAG2
                 LOOP
           ENDCASE
        ENDDO

         USE TEMP2

         ? "                           PROMOTION HISTORY FOR :"
         ? "   "
         LIST "      " + SERNO + "   " + NAME + "   " + RANKDES +;
 "  " + CORPSDES OFF
         ? "   "
         ? " RANK                    DATE        MONTH       YEAR;
          ORDER-NUMBER"
         ? " ==================================================;
=============================="

         USE TF4
         LIST " " + RANKDES + "  " + RANKDD + "          " + ;
RANKMM + "         " + RANKYY + "        " + RORNO OFF
          SET COLOR TO 4
         ? " ==================================================;
=============================="
         ? "                         TYPE ANY KEY TO CONTINUE"
         SET CONSOLE OFF
         WAIT
         SET CONSOLE ON
         SET PRINT OFF
      ENDIF
    ENDDO
    ERASE
ENDDO
ERASE
RETURN
```

```
*****************************************************************
* PROGRAM NAME : QUERYD.PRG                                     *
* AUTHOR       : DJOKO M. ARIYADI                               *
* DATE WRITTEN : MAY, 1986                                      *
* PURPOSE      : TO OBTAIN AN OFFICER'S JOB HISTORY, GIVEN A    *
*                SERIAL NUMBER                                  *
* PROGRAMS CALLED BY THIS PROGRAM ARE :                         *
* DBF FILES USED ARE : OFFICER, CORPS, RANKLIST, JOBOFF, JOBLIST*
*****************************************************************
SET TALK OFF
ERASE
SET CONSOLE ON
STORE T TO FLAG
SET COLOR TO 3Ø,14
STORE T TO Q:FLAG
DO WHILE Q:FLAG
    ERASE
    ? "   "
    @ 2,1 SAY "QUERY                       OFFICER'S JOB;
 HISTORY                    MAY 1986"

    SET COLOR TO 4
    ? "=========================================================;
======================="
    SET COLOR TO 3Ø,14
    ? "   "
    ? "   "
    ? "   "
    ? "   "
    ? "       Given a Officer's Serial Number, This program;
respond with : "
    ? "   "
    ? "        OFFICER'S JOB HISTORY, INCLUDES RANK, DATE,;
 STATION, AND CITY"
    ? "   "
    ? "   "
    ? "      The following is a list of Officer's Serial;
 Number and their name "
    ? "   "
    ? "        You can query one serial number at a time;
 to know"
    ? "   "
    ? "   "
    ? "   "
    ? "   "
    SET COLOR TO 4
    ?"==========================================================;
======================="
    ? "   "
    ? "                    TYPE ANY KEY TO CONTINUE"
    SET CONSOLE OFF
    WAIT
    SET CONSOLE ON
    ERASE
    DO WHILE Q:FLAG
      ERASE
    @ 2,1 SAY "QUERY                       OFFICER'S JOB HISTORY;
                    MAY 1986"
      SET COLOR TO 4
      ? "=======================================================;
=========================="
      SET COLOR TO 30,14
      USE OFFICER
      INDEX ON SERNO TO DXOFF
      SET INDEX TO DXOFF
      ? "          ,     OFFICER'S SERIAL #, ALREADY ON THIS;
 FILE"
      ?  "  "
      ?  "                         SERIAL #      NAME"
      ?  "  "
      LIST "                          " + SERNO + "     " + ;
NAME OFF
      SET COLOR TO 30,14
      ?  "  "
      ?  "  "
      TEXT
         PLEASE SELECT ONE DESIRED OFFICER' SERIAL NUMBER,;
 ACCORDING TO
        THE ABOVE LIST (6 DIGITS PLEASE) e.g. "001111"  AND ;
HIT "RETURN" KEY
      ENDTEXT
      SET COLOR TO 4
      ?  "  "
      ?  "             TO FINISH (RETURN TO QUERY MENU OPTION);
 HIT 'RETURN'"
      ?  "  "
      SET COLOR TO 30,14
      ACCEPT "                     ENTER SERIAL NUMBER e.g;
 ( 6 DIGITS ) " TO Q:MNO

      ERASE
      @ 12,20 SAY "BE PATIENT PLEASE, STILL PROCESSING"
      IF Q:MNO = " "
        RELEASE ALL LIKE Q*
        ERASE
        STORE F TO Q:FOUND
        STORE F TO Q:FLAG
      ELSE
        STORE F TO Q:FOUND
        FIND "&Q:MNO"
      IF #=0
         ERASE
         SET COLOR TO 112,140
         @ 10,9 SAY "                SORRY, SERIAL NUMBER CANNOT BE;
 FOUND"
         ?  "  "
         ?"              PLEASE, SEE THE SERIAL NUMBER LIST;
 AND TRY AGAIN"
         ?  "  "
         ?"                         TYPE ANY KEY, TO CONTINUE"
         SET CONSOLE OFF
         WAIT
         SET CONSOLE ON
         ERASE
         STORE F TO Q:FOUND
      ELSE
         STORE T TO Q:FOUND
      ENDIF,
      ENDIF
      IF Q:FOUND
         STORE NAME TO Q:OFF
         ERASE
         STORE SERNO TC Q:MNO
         USE OFFICER
         COPY TO TF1 FOR SERNO = "&Q:MNO"

         USE TF1
         SELECT SECONDARY
         USE RANKLIST
   JOIN TC TEMP1 FOR P.RANKID=S.RANKID FIELDS P.SERNO,P.NAME,;
RANKDES,P.CORPS

         USE TEMP1
         SELECT SECONDARY
         USE CORPCODE
   JOIN TO TEMP2 FOR P.CORPS=S.CORPS FIELDS P.SERNO,P.NAME,;
P.RANKDES,CORPSDES

         USE TF1
         SELECT SECONDARY
         USE JOBOFF
  JOIN TO TF FOR P.SERNO=S.SERNC FIELDS JOBRANK,JOBID,JOBDD,;
JOBMM,JOBYY

         USE TF
         SORT ON JOBRANK TO TF2 DESCENDING

         USE TF2
         SELECT SECONDARY
         USE RANKLIST
  JOIN TO TF3 FOR P.JOBRANK=S.RANKID FIELDS RANKDES,P.JOBID,;
P.JOBDD,P.JOBMM,P.JOBYY

         USE TF3
         SELECT SECONDARY
         USE JOBLIST
  JOIN TO TF4 FOR P.JOBID=S.JOBID FIELDS JOBDES,P.RANKDES,;
P.JOBDD,P.JOBMM,P.JOBYY,JOBSTAT

         USE TF4
         SELECT SECONDARY
         USE STATLIST
  JOIN TO TF5 FOR P.JOBSTAT=S.STATCODE FIELDS CITYSTAT,STATDES,;
P.JOBDES,P.RANKDES,P.JOBDD,P.JOBMM,P.JOBYY

         USE TF5
         SELECT SECONDARY
         USE CITYLIST
  JOIN TO TF6 FOR P.CITYSTAT=S.CITYCODE FIELDS P.RANKDES,;
P.JOBDES,P.JOBDD,P.JOBMM,P.JOBYY,P.STATDES,CITYDES

         STORE T TO Q:FLAG2
         DO WHILE Q:FLAG2
         @ 10,8 SAY "     DO YOU WANT TO SEE THE LIST ON SCREEN;
 or HARDCOPY ?"
         ?"  "
         ACCEPT "                 ENTER 's' FOR SCREEN or 'h';
 FOR HARDCOPY" TO Q:PRN
         READ
           DO CASE
              CASE Q:PRN = 's'
                 STORE F TO Q:FLAG2
                 ERASE
              CASE Q:PRN = 'h'
                 STORE F TO Q:FLAG2
                 ERASE
                 SET PRINT ON
                 ERASE
              OTHERWISE
                 DO ERROR
                 ERASE
                 STORE T TO Q:FLAG2
                 LOOP
           ENDCASE
        ENDDO

         USE TEMP2
         ? "                              JOB HISTORY FOR :"
         ? "  "
         LIST "      " + SERNO + "   " + NAME + "  " + RANKDES +;
 " " + CORPSDES OFF
         ? "  "
         ? "     RANK               JOB                          ;
 MM/YY    STATION    CITY"
         ? "==================================================;
=============================="

         USE TF6
         INDEX ON JOBYY+JOBMM TO TF7
         USE TF6 INDEX TF7
         LIST RANKDES,JCBDES,JOBMM,JOBYY,STATDES,CITYDES OFF
          SET COLCR TO 4
         ? " ===================================================;
=============================="
         ? "                         TYPE ANY KEY TO CONTINUE"
         SET CONSOLE OFF
         WAIT
         SET CONSOLE ON
         SET PRINT OFF
      ENDIF
    ENDDO
    ERASE
ENDDO
ERASE
RETURN
```

```
*****************************************************************
* PROGRAM NAME : QUERYE.PRG                                     *
* AUTHOR       : DJOKO M. ARIYADI                               *
* DATE WRITTEN : MAY, 1986                                      *
* PURPOSE      : TO OBTAIN AN OFFICER'S JOB ELIGIBLE            *
* PROGRAMS CALLED BY THIS PROGRAM ARE :                         *
* DBF FILES USED ARE : OFFICER, CORPS, RANKLIST, JOBOFF, JOBLIST*
*****************************************************************
SET TALK OFF
ERASE
SET CONSOLE ON
STORE T TO FLAG

SET COLOR TO 3Ø,14
STORE T TO Q:FLAG
DO WHILE Q:FLAG
    ERASE
    ? "  "

    @ 2,1 SAY "QUERY                     OFFICER'S JOB ;
ELIGIBLE                MAY 1986"

    SET COLOR TO 4
    ? "=====================================================;
======================="
    SET COLOR TO 3Ø,14
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ? "        Given a Officer's Serial Number,this program ;
respond with: "
    ? "  "
    ? "       JOB ELIGIBLE INCLUDES, JOB-ID, DESCRIPTION, ;
STATION AND CITY"
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    ? "  "
    SET COLOR TO 4
    ?"======================================================;
======================"
    ? "  "
    ? "                      TYPE ANY KEY TO CONTINUE"
    SET CONSOLE OFF
    WAIT
    SET CONSOLE ON
    ERASE
    DO WHILE Q:FLAG
      ERASE
    @ 2,1 SAY "QUERY                   OFFICER'S JOB ELIGIBLE;
                     MAY 1986"
      SET COLOR TO 4
      ? "==========================================================;
==========================="
      SET COLOR TO 30,14
      ? "  "
      ? "  "
      ? "  "
      ? "  "
      ? "  "
      ? "  "
      TEXT
             PLEASE SELECT ONE DESIRED OFFICER' SERIAL NUMBER,;
 WHICH YOU

         LIKE TO KNOW (6 DIGITS PLEASE) e.g. "333333"  AND HIT ;
"RETURN" KEY
      ENDTEXT
      SET COLOR TO 4
      ? "  "
      ? "  "
      ? "               TO FINISH (RETURN TO QUERY MENU OPTION);
 HIT 'RETURN'"
      ? "  "
      SET COLOR TO 30,14
      ? "  "
      ? "  "
      ACCEPT "                          ENTER SERIAL NUMBER e.g:
 ( 6 DIGITS ) " TO Q:MNO

      ERASE
      @ 12,20 SAY "BE PATIENT PLEASE, STILL PROCESSING"
      USE OFFICER
      INDEX ON SERNO TO DXOFFS
      USE OFFICER INDEX DXOFFS
      IF Q:MNO = " "
        RELEASE ALL LIKE Q*
        ERASE
        STORE F TO Q:FOUND
        STORE F TO Q:FLAG
      ELSE
        STORE F TO Q:FOUND
        FIND "&Q:MNO"
      IF #=0
         ERASE
         SET COLOR TO 112,140
         @ 10,9 SAY "                SORRY, SERIAL NUMBER CANNOT BE;
 FOUND"
         ? "  "
        ?"                  PLEASE, SEE THE SERIAL NUMBER LIST AND;
 TRY AGAIN"
         ? "  "
         ?"                          TYPE ANY KEY, TO CONTINUE"
         SET CONSOLE OFF
         WAIT
         SET CONSOLE ON
         ERASE
         STORE F TO Q:FOUND
      ELSE
         STORE T TO Q:FOUND
      ENDIF
      ENDIF
      IF Q:FOUND
         STORE NAME TO Q:OFF
         ERASE
         STORE SERNO TO Q:MNO
         USE OFFICER
         COPY TO TF15 FOR SERNO = "&Q:MNO"

         USE TF15
         SELECT SECONDARY
         USE RANKLIST
   JOIN TO TEMP15 FOR P.RANKID=S.RANKID FIELDS P.SERNC,P.NAME,;
RANKDES,P.CORPS

         USE TEMP15
         SELECT SECONDARY
         USE CORPCODE
   JOIN TO TEMP25 FOR P.CORPS=S.CORPS FIELDS P.SERNO,P.NAME,;
P.RANKDES,CORPSDES

         STORE T TO Q:FLAG3
         DO WHILE Q:FLAG3
         @ 10,8 SAY "     JOB AVAILABLE FOR THE SAME RANK or ;
AFTER PROMOTION ?"
         ?"  "
         ACCEPT "                  ENTER 's' FOR SAME RANK or 'p';
 PROMOTION" TO Q:PRN
         READ
           DO CASE
              CASE Q:PRN = 's'
                 STORE F TO Q:FLAG3
                 use tf15
                 SELECT SECONDARY
                 USE JOBRNK
                 JOIN TO TF25 FOR P.RANKID=S.JOBRK FIELDS JOBID
                 ERASE
              CASE Q:PRN = 'p'
                 STORE F TO Q:FLAG3
                 ERASE
                 USE TF15
                 STORE 1 TO RK
                 STORE RANKID TO MRANK
                 STORE MRANK+RK TO RANKID
                 USE JOBRNK
                 COPY TO TF25 FOR JOBRK = RANKID
              OTHERWISE
                 DO ERROR
                 ERASE
                 STORE T TO Q:FLAG3
                 LOOP
           ENDCASE
         ENDDO

         USE TF15
         SELECT SECONDARY
         USE JOBEDU
  JOIN TO TF35 FOR P.EDUID=S.JOBED FIELDS JOBID

         USE TF15
         SELECT SECONDARY
         USE JOBCORP
  JOIN TO TF45 FOR P.CORPS=S.JOBCO FIELDS JOBID

         USE TF25
         SELECT SECONDARY
         USE TF35
  JOIN TO TF55 FOR P.JOBID=S.JOBID FIELDS JOBID

         USE TF55
         SELECT SECONDARY
         USE TF45
  JOIN TO TF65 FOR P.JOBID=S.JOBID FIELDS JOBID

         USE JOBLIST
         SELECT SECONDARY
         USE TF65
  JOIN TO TF75 FOR P.JOBID=S.JOBID FIELDS P.JOBID,JOBDES,JOBSTAT

         USE TF75
         SELECT SECONDARY
         USE STATLIST
  JOIN TO TF85 FOR P.JOBSTAT=S.STATCODE FIELDS STATDES,CITYSTAT,;
P.JOBID,P.JOBDES

         USE TF85
         SELECT SECONDARY
         USE CITYLIST
  JOIN TO TF95 FOR P.CITYSTAT=S.CITYCODE FIELDS P.JOBID,;
P.JOBDES,P.STATDES,CITYDES

         STORE T TO Q:FLAG2
         DO WHILE Q:FLAG2
         @ 10,8 SAY "     DO YOU WANT TO SEE THE LIST ON SCREEN;
 or HARDCOPY ?"
         ?"  "
         ACCEPT "                 ENTER 's' FOR SCREEN or 'h';
 FOR HARDCOPY" TO Q:PRN
         READ
           DO CASE
              CASE Q:PRN = 's'
                 STORE F TO Q:FLAG2
                 ERASE
              CASE Q:PRN = 'h'
                 STORE F TO Q:FLAG2
                 ERASE
                 SET PRINT ON
                 ERASE
              OTHERWISE
                 DO ERROR
                 ERASE
                 STORE T TO Q:FLAG2
                 LOOP
           ENDCASE

         USE TEMP25

         ? "                           JOBS ELIGIBLE FOR :"
         ? "  "
         LIST "       " + SERNO + "    " + NAME + "   " + RANKDES;
 + "  " + CORPSDES OFF
         ? "  "
         ?  "  JOB-ID         DESCRIPTION                    STATION
;  CITY"
         ? " ===================================================;
============================="

         USE TF95
         LIST "   " + JOBID + "    " + JOBDES + "   " + STATDES + ;
"    " + CITYDES OFF
          SET COLOR TO 4
         ? " ===================================================;
============================="
         ? "                           TYPE ANY KEY TO CONTINUE"
         SET CONSOLE OFF
         WAIT
         SET CONSOLE ON
         SET PRINT OFF
      ENDIF
    ENDDO
    ERASE
ENDDO
ERASE
RETURN
```

LIST OF REFERENCES

1. Mabes TNI-AL, Sistem_Personil_TNI-AL_Serta_Pola_Umum Pembinaannya_(UM-1.03-A), pp. 10-13, Skep Kasal No: Skep/2874/XI/ 1983, Tanggal 18 Nopember 1983.

2. Boehm, B.W., Software_Engineering_Economic, pp. 35-37 Prentice Hall, 1981.

3. Kroenke, David M .,Database_Processing, p. 11, Second edition, Science Research Associates, Inc 1983, 1977.

4. Gorman Michael M., Managing___Data___Base____For Critical_Factors, 1984 QED Information Sciences, Inc.

5. Martin James., System____Design____From____Provably Correct Constracts, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1985.

BIBLIOGRAPHY

1. Byers, Robert A ., Everyman's__Database___Primer Featuring_dBase_II, Ashon-Tate, 1982.

2. Castro Luis, Hanson Jay, Rettig Tom., Advanced Programmers_Guides._Featuring_dBase_III_and_dBase_II, Ashton-Tate, 1985.

3. Hannan James., Database___Management., Auerbach Publishers Inc, 1982.

4. Jardine Donald A., Database__Management__Systems, North-Holland Publishing Company., 1974

5. Jones Edward., Using_Dbase_III, Obsborne McGraw-Hill, 1985.

6. Lee, Sang M and Carry D. Thorp, Jr., Personnel Management_A_Computer-Based_System, Petrocelli Books, Inc, 1978.

7. Martin James., Design__And__Strategy__For_Distributed Data Processing, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1981.

8. McFarland, Dalton E ., Personnel__Management__Theory and_Practice, The Macmillan Company, 1968.

9. Pigors Paul and Myers Charles., Personnel Administration___A___Point___Of__View__And__A__Method, McGraw-Hill, 1969.

10. Wiederhold, Gio., Database__Design, McGraw-Hill Inc, 1983.

11. Williams, Martha E., Annual__Review__of___Information Science and Technology Vol 14 1979, Knowledge Industry Publications,Inc, 1979.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22314

2. Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93943-5000

3. Curricular Officer, Code37 1
Computer Technology
Naval Postgraduate School
Monterey, California 93943-5000

3. Department Chairman, Code 54 1
Deparment of Administrative Sciences
Naval Postgraduate School
Monterey, California 93943-5000

4. Prof. Daniel R. Dolk PhD 1
Code 54DK
Naval Postgraduate School
Monterey, California 93943-5000

5. Prof. R. A. McGonigal PhD 1
Code 54MB
Naval Postgraduate School
Monterey, California 93943-5000

6. Komandan Korps Marinir 1
Jln. Kwitang
Jakarta Pusat, Indonesia

7. Deputy Personil Kasal 1
Mabes TNI-AL
Cilangkap,
Jakarta, Indonesia

8. Kepala Biro Pullahta Setjen Dephankam 1
Jl. RS Fatmawati - Pondok Labu,
Jakarta Selatan, Indonesia

9. Kadis Pullahta TNI-AL 1
Wisma Lumba-Lumba
Jl. Gatot Subroto 12 - Senayan
Jakarta Pusat, Indonesia

10. Ibu Darmadi 1
Jl. Dua No. 39 Perum Korpri Kramat
Magelang 56115, Indonesia.

11. Major Mar. Djoko Murti Ariyadi 4
Komplex Biro Pullahta Hankam No. G-1
Jl. RS Fatmawati - Pondok Labu,
Jakarta Selatan, Indonesia